An Argus Specialist Publication

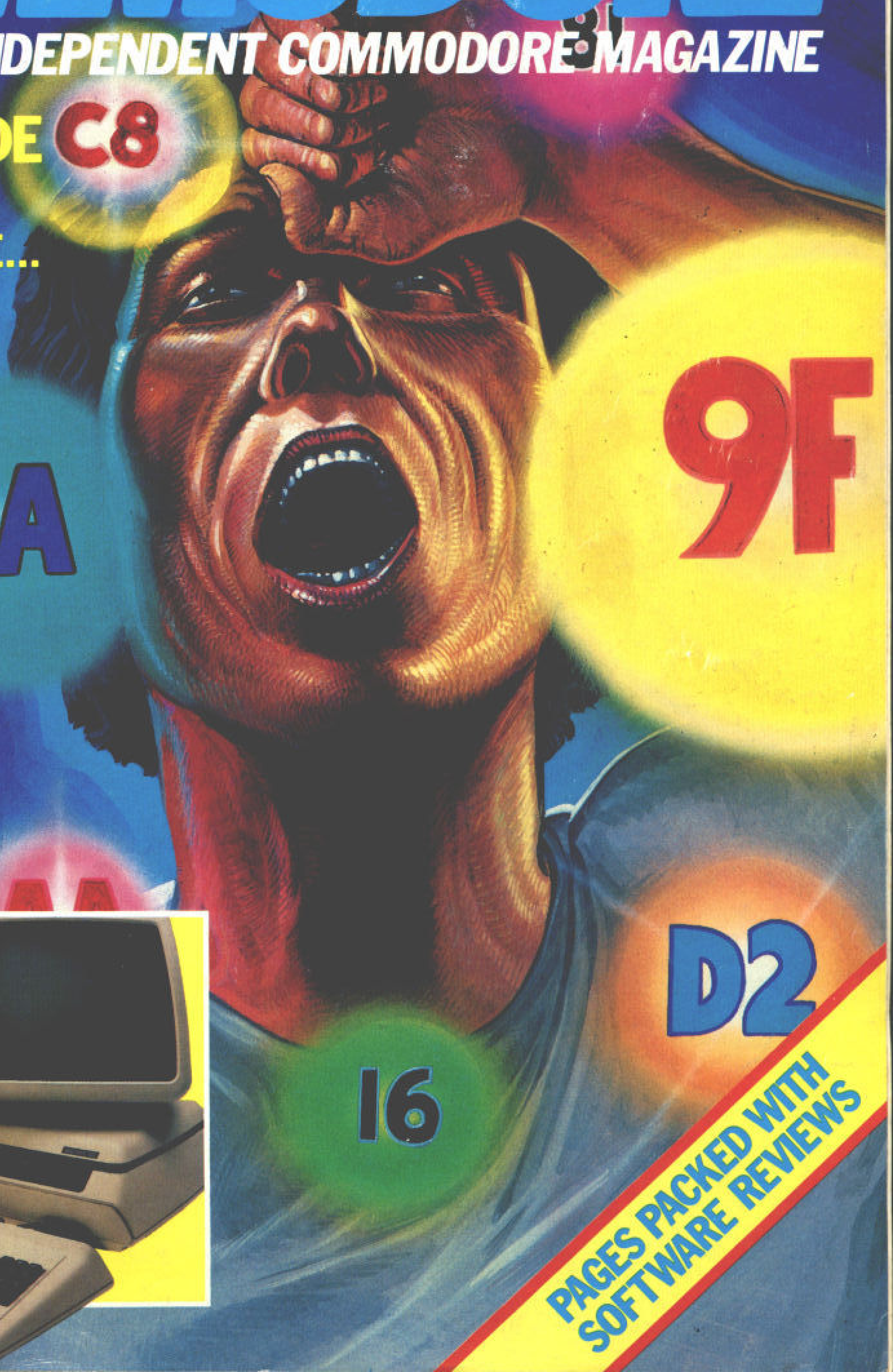
OCTOBER 1984

80p

NEW

Your COMMODORE

YOUR BEST INDEPENDENT COMMODORE MAGAZINE



MACHINE CODE TOO MUCH?
FIND RELIEF INSIDE...

LEARN ABOUT BASIC WITH OUR GREAT IN-DEPTH SERIES

GAMES AND UTILITIES TO TYPE IN

ADDICTED TO ADVENTURE? READ TALES FROM THE CRYPT



DOWN TO BUSINESS:
CBM8296 HARDWARE REVIEW

PAGES PACKED WITH SOFTWARE REVIEWS

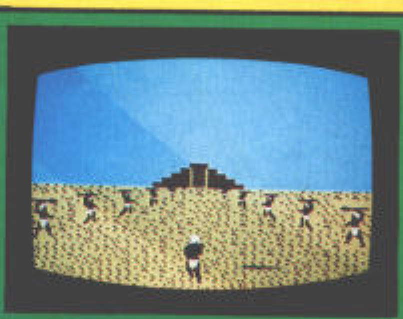
DON'T JUST SIT THERE - PLAY SOMETHING!

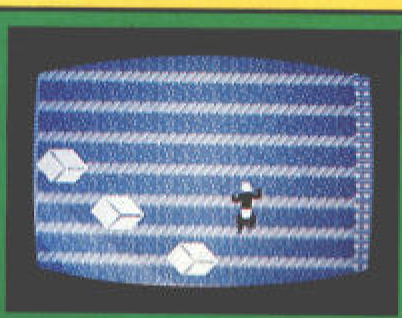
FORBIDDEN FOREST

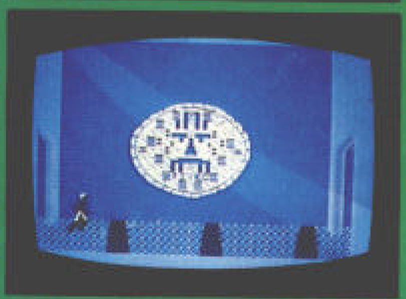
Forbidden Forest is more of a quest than just a game! The action takes place in a four dimensional scrolling forest landscape which many have entered, but none has returned. Yes, I did say FOUR dimensional - day fades into night as the action unfolds! The quest is to seek out and destroy the Demogorgon, mystic ruler of the Forbidden Forest. Before you can even set eyes on him you will have to contend with his army of fearsome creatures, including mutant spiders, showers of giant frogs, snakes, dragons, skeleton soldiers and more! You have only your trusty bow and arrows to depend on!



SS018







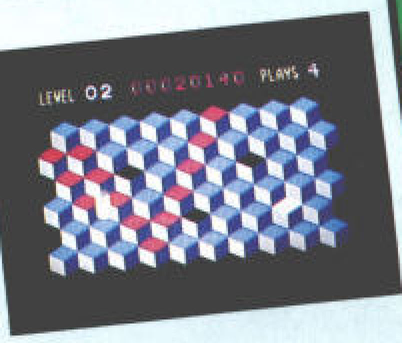
AZTEC CHALLENGE

A challenge on an epic scale! Aztec Challenge takes you on a journey to Mexico and the ancient pyramid of Tenochtitlan. The ancient Aztec gods and their devotees have ensured that no ordinary human can learn the secrets of the temple and live to tell the tale. The pyramid is protected by all manner of treacherous traps and hidden perils - an epic test of your courage and cunning. Aztec Challenge features no less than seven totally different screens - here are just three of them - each of which presents a brand new challenge. We hope your joystick can stand up to it!

SS019

SLINKY

Slinky, the spring, was having fun hopping about when suddenly he came upon a pile of coloured blocks, so he thought he'd play around on them for a while. Much to his amazement he found that they changed colour when he landed on them. Wow! But unknown to him, the blocks belonged to the Wicked Wizard, who sent his friends along to tease our poor hero. Slinky is a real fun package with ninety-nine levels, amazing reward displays, and action replays. Where else could you meet such charming characters as Dusty the dust cloud, Marge the magnet, Ralph the random raindrop, and Lorenzo the chameleon hopper?



SS020

ON CASSETTE £8.95
ON DISK £12.95

FOR THE commodore 64

Audiogenic LTD

P.O. BOX 88. READING, BERKS.

SEND FOR FREE COLOUR CATALOGUE!

Our COMMENT

Our Editor has taken time out from the daily drudgery of journalistic life (and generally enjoying himself) to introduce his new magazine.



CONGRATULATIONS!
You have had the good sense and judgment to pick up a copy of the first issue of a GREAT new magazine dedicated to the Commodore range of microcomputers. If you have actually purchased this copy and are now sitting in your armchair at home you can rest assured in the knowledge that this and future issues of Your Commodore will satisfy your thirst for information, games, serious software, education, news and all sorts of goodies that are part and parcel of the Commodore scene. On the other hand if you haven't already parted with money to buy this copy — Why not? We can assure you here and now that it will be money well spent — you'll gain an invaluable insight into your Commodore micro and what it can do for you!

What can we offer?

Assuming that you have spent a couple of minutes flicking through this magazine and you are still not convinced that Your Commodore is the best thing since sliced silicon, spend a little more time in my company and let me try to change your mind. . .

Your Commodore will entertain, inform and educate you on all matters 'a la Commodore'. Each issue will have regular news pages to keep you informed of all the latest products and stories related to your computer; our intrepid software reviewers will be let loose each month on the latest packages around and let you in on their opinions before you actually hand over some cash; Runecaster will be summoned from the Crypt to advocate the spirit of Adventure; and at least one major piece of hardware will be reviewed each issue.

Commodore are not exactly renowned for the quality and clarity of their manuals — we have series and articles on programming in both BASIC and machine code to help you write programs yourself and to help you understand the way other programs are written — whether you are a beginner or expert there's sure to be something here for you to learn!

For the average home computer user (although we of course accept that Your Commodore readers will be well above 'average'!), games occupy a tremendous amount of the time and energy spent on the computer. Your Commodore will cater for the games player, as evidenced by our feature on games programming on the VIC and the fantastic games for the VIC and Commodore 64 in this issue. However we are trying not to go 'games-mad' and we appreciate that there are an awful lot (sorry, not meant literally!) of users out there in Commodore-Land who have exhausted their trigger fingers and crashed all their spaceships!

Seriously, though. . .

In Your Commodore we have put together a variety of articles for the more serious-minded amongst our readers: there are some really useful routines that are primarily intended for the Commodore 64, although they will run on other Commodore micros with a few alterations. The business users of Commodore micros have not been forgotten, either. Each issue of Your Commodore will have a number of pages devoted to this growing and generally under-served application area, — under the highly original title of 'Down to Business'! As you can see in this copy, we have a review of the CBM 8296-D and we take a look at how to set about writing your own business-type software.

Not only. . .

It would be true to say that we could go on ad infinitum about the value, expertise and entertainment that you are going to get from Your Commodore, but. . .

All you really need to know is that if you have a Commodore micro (regardless of which model) and you want to keep informed on the latest happenings in the world of Commodore, then just buy, read and inwardly digest each fantastic issue of Your Commodore.

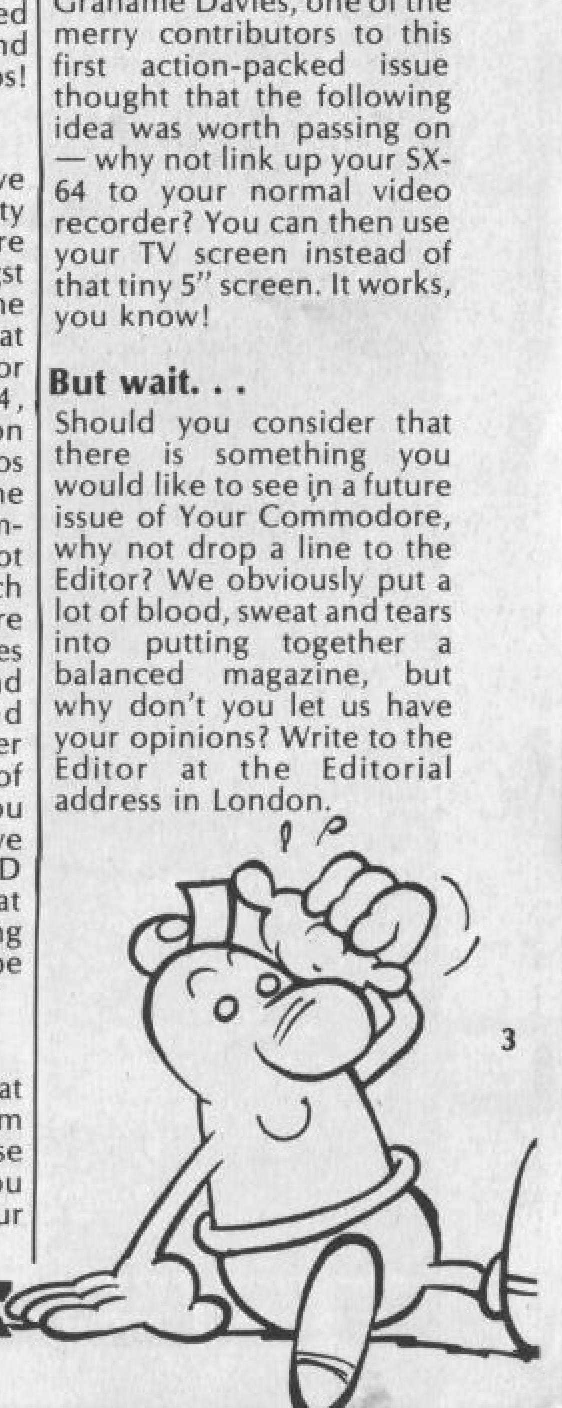
Now pay your money and enter the fascinating world of Your Commodore!

Passing thought

Grahame Davies, one of the merry contributors to this first action-packed issue thought that the following idea was worth passing on — why not link up your SX-64 to your normal video recorder? You can then use your TV screen instead of that tiny 5" screen. It works, you know!

But wait. . .

Should you consider that there is something you would like to see in a future issue of Your Commodore, why not drop a line to the Editor? We obviously put a lot of blood, sweat and tears into putting together a balanced magazine, but why don't you let us have your opinions? Write to the Editor at the Editorial address in London.





VOLUME 1
NUMBER 1
OCTOBER 1984
Launch Editor: Wendy J Palmer
Advertisement Manager: Mike Segrue
Advertisement Copy Control: Sue Couchman
Managing Editor: Ron Harris
Chairman: Jim Connell
Origination: Ebony Typesetting
Design: MM Design

Editorial & Advertisement Office
No 1 Golden Square,
London W1R 3AB
Telephone: 01-437 0626
Telex: 8811896

Your Commodore is a monthly magazine appearing on the first Friday of each month.

Distribution by: Argus Press Sales & Distribution Ltd, 12-18 Paul Street, London EC2A 4JS.
Printed by: Alabaster Passmore & Sons Ltd, Tovil, Maidstone, Kent.

Subscription rates upon application to Your Commodore Subscriptions Department, Infonet Ltd, Times House, 179 The Marlowes, Hemel Hempstead, Herts. HP1 1BB.
The contents of this publication including all articles, designs, plans, drawings and programs and all copyright and other intellectual property rights therein belong to Argus Specialist Publications Limited. All rights conferred by the Law of Copyright and other intellectual property rights and by virtue of international copyright conventions are specifically reserved to Argus Specialist Publications Limited and any reproduction requires the prior written consent of the Company. ©1984 Argus Specialist Publications Limited.

MASTERING MACHINE CODE 6

The mysteries of machine code are explained in this tremendous series.

BATTLE ATAK 10

The sirens are sounding and you are at war! Can you defeat the enemy before he gets you?

MACHINE CODE TO BASIC 13

A great little utility that does exactly what you would think — converts machine code to BASIC!

BASIC FACTS PART 1 14

BASIC is supposed to be such a simple language but few of us know and understand everything. We start at the beginning. . .

FAST TAPE SEARCH 28

A great way of speeding up the Commodore tape system.

SOFTWARE SPOTLIGHT 30

Our fearless reviewers have been let loose on some of the latest software around for the Commodore 64 and VIC 20. So read on before you spend out some money. . .

YOUR OWN ARCADE GAME 40

So you think you can do better than the games you buy? We give you some general guidelines on writing your own!

CONCORDE II 43

Fasten your seatbelts and prepare to land the fastest aircraft in the world — Concorde II.



ARTISTIC STUFF 17

Fancy your chances as an artist? Our reviewer takes a look at a package that's almost as useful as a new paint brush.

CODE BREAKER 18

Do you consider yourself endowed with a logical mind? See if you can beat the computer at a great game of deduction and guesswork!

RENUMBERING 22

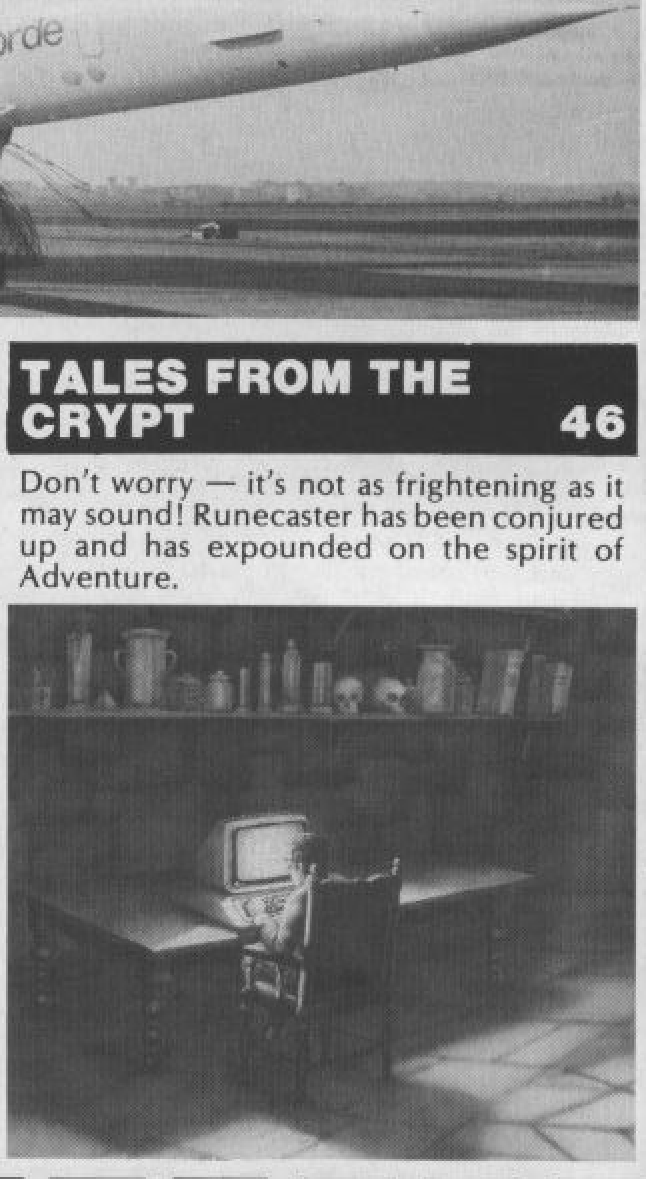
If your programs end up looking like a piece of numerical spaghetti, this useful routine could be just what you need.

SPRITE DESIGNER 24

Be spritely and do fantastic things with your Commodore 64.

TALES FROM THE CRYPT 46

Don't worry — it's not as frightening as it may sound! Runecaster has been conjured up and has expounded on the spirit of Adventure.



CONTENTS



INTRODUCING MODEMS 48

Modems have become rather fashionable in computing circles just lately. . .we let you into the secrets of their appeal.

MAKING MUSIC 50

You are chord-ially (!?) invited to make learning music that much easier with the use of your Commodore 64.

INPUT/OUTPUT 52

Some commonly asked technical enquiries answered in easy-to-understand language.

TRACK KING 54

Drive yourself crazy with this racey game for the VIC 20.

LADDER MAZE 56

Can you climb to great heights and get lost in the intricacies of this VIC 20 game?

REFERENCE LIBRARY 58

We've taken some of the Commodore books off our shelves and let you know whether they're worth reading.



VIC GAMES PROGRAMMING 61

We know there are thousands of VIC gamers out there and here we help you to write your own games.

COMMODORE 64 EXPOSED 64

We've taken more than a leaf out of this great book from Melbourne House and give you a taster of things to come from within the covers.

RELATIVE FILES 70

Relative files can be somewhat of a problem for some of you — but be confused no longer!

DATA STATEMENTS 73

All the news and information for Commodore owners.

DOWN TO BUSINESS

DOING IT YOURSELF 80

You shouldn't be put off writing your own business software — it's certainly isn't impossible and we try to show you how.

IS BIGGER BETTER? 82

The CBM 8296-D is an attractive looking machine. Does its performance match its appearance?

SUPERBASE OVERVIEW 86

Precision's Superbase has received a lot of exposure — what does our 'expert' think of it?

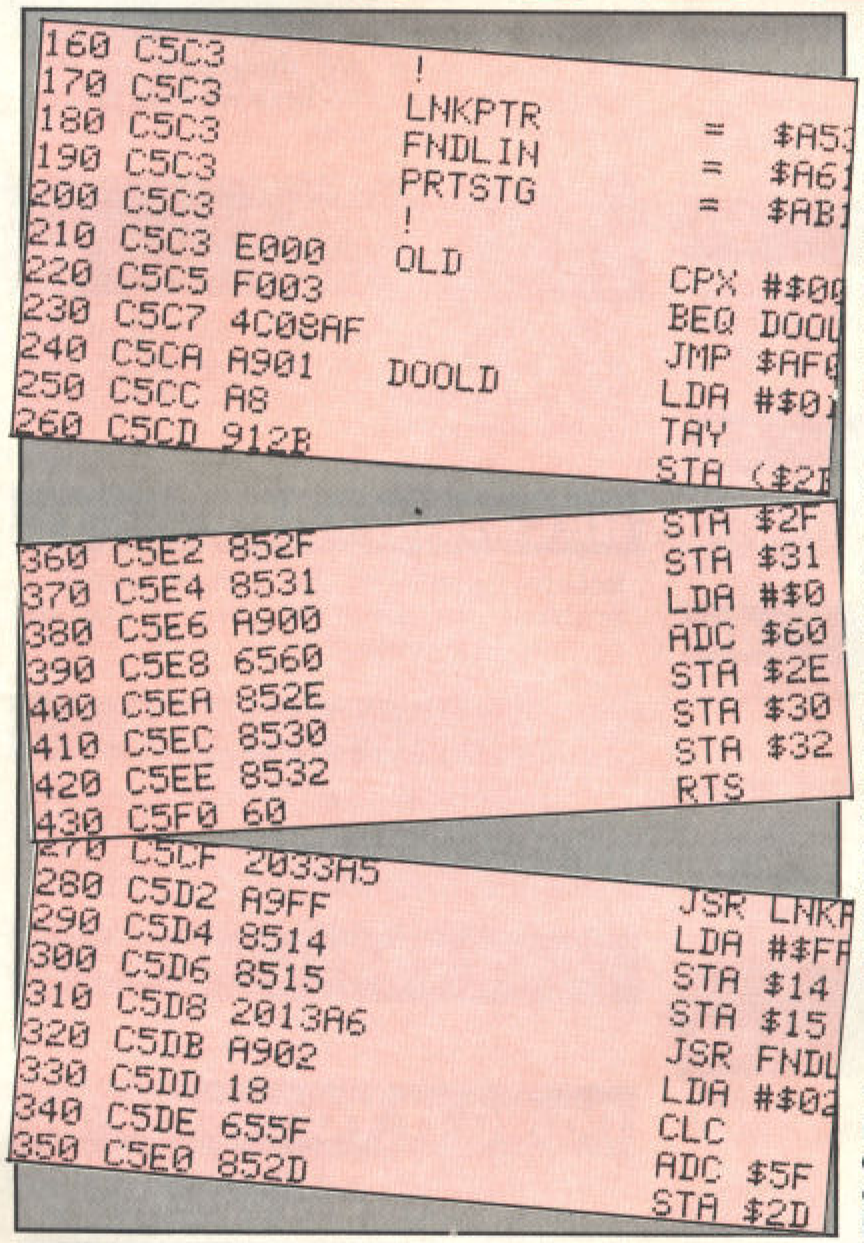
MASTERING MACHINE CODE

Machine code is the lowest-level computer language but it is also the fastest. A P and D J Stephenson dispel some of its mysteries.

NEWCOMERS TO THE ART must be forgiven for thinking that BASIC is somehow the natural language of the home or business microcomputer. The written information supplied with machines, such as the Commodore 64 User Manual, tend to give this impression. BASIC is certainly the most popular and the easier to learn but it is a mistake to think of it as the natural language of the Commodore 64.

In fact, BASIC is not natural to any computer. This honour is claimed by a language, if indeed it can be called a true language, known (loosely) as **machine code**. The **microprocessor** chip, which functions as the central control of a computer, can only respond to orders couched in machine code. Even when we write a program in BASIC, a complex program, called an **interpreter,** is working feverishly in the background, converting each line of the program into machine code ready for the attention of the microprocessor.

An obvious question now arises — if machine code is the natural language of a computer, why is BASIC so popular? The question is best answered by comparing their relative advantages and disadvantages.



•BASIC is easier to learn, to use, than machine code.
• Programs written in BASIC are slow to execute. Machine code programs execute somewhere between a hundred and a thousand times faster than the same program written in BASIC.
• A machine code program occupies less memory space than the equivalent BASIC version.
• Learning machine code forces you to understand the inner workings of a computer. However clever you become with BASIC, the computer itself will still remain a mysterious grey box.
• Skill in machine code tends to inspire awe at the local computer club. (Whether this is considered trivial or not depends on your temperament.)

It is clear that machine code scores on all counts except simplicity.

How Hard Is Machine Code?

We have said that machine code is not simple, but this does not necessarily mean you will find it particularly hard. It requires just that little more dedication and patience than BASIC requires, a different attitude of mind and, above all, a greater attention to detail. The language is more abstract and code-like in form. The example below illustrates the code-like form. It adds 3 and 5 together and stores the result in a machine address.

Assembly code	Pure machine code
LDA #00	A9 00
ADC #3	69 03
ADC #5	69 05
STA $4000	8D 00 40

The lines on the left are written in assembly code which is a superior type of machine code. (The difference is explained later.) The version on the right is the same program but written in pure machine code. In BASIC, the same effect can be achieved with, say,

```
Z=3+5
```

You will have to understand **binary** arithmetic**, hexadecimal** code, certain **logical** operations and a few other bits of background knowledge. But none of these subjects need frighten you. They are not hard to learn and the rewards are well worth the effort. However, when you do become proficient, don't adopt a snobbish attitude towards BASIC. BASIC is a nice language and should not be though of an a competitor of machine code. In fact, the aim of this series is directed towards the amalgamation of BASIC with machine code.

Although it was stated above that machine code is much faster than BASIC, it would be quite wrong to attach too much significance to it. If a particular program runs fast enough written in BASIC, there is little point in re-writing it in machine code. On the other hand, there may be certain parts of a BASIC program which run far too slow for comfort.

For example, sorting a long list of data into order can take several minutes or, in some cases, even hours in BASIC. A machine code **subroutine** could achieve the same result in a few seconds. Screen animation effects are far better in machine code than in BASIC. The movements are less jerky and the display can be given a more professional appearance in machine code. Robotics is becoming popular with computer hobbyists and machine code control of the robot's private parts is virtually a necessity — BASIC is seldom fast enough.

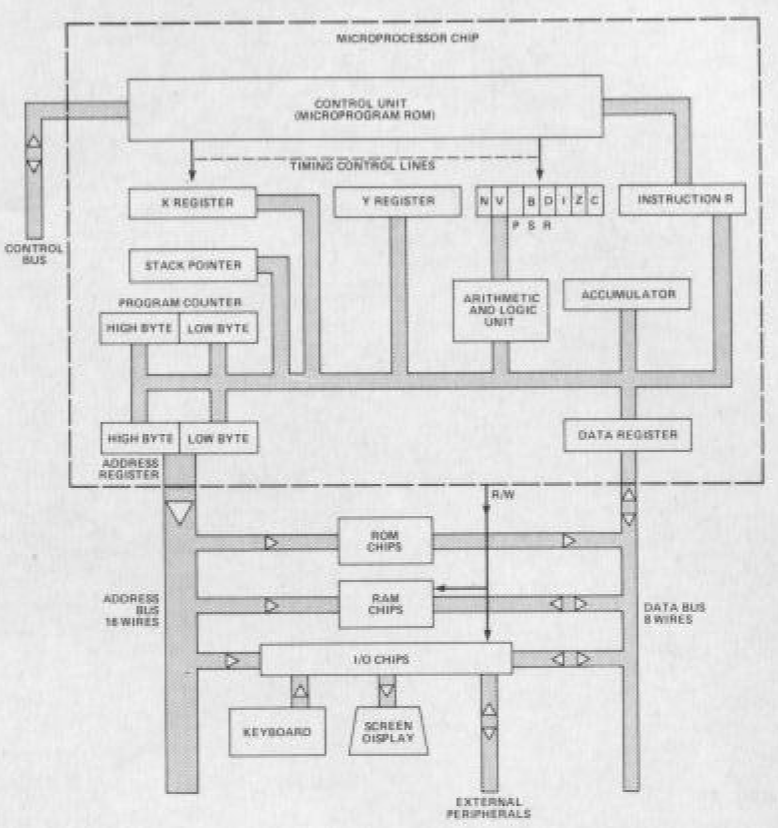


Fig. 1

Entering Machine Code

There are two methods — one is free but tiresome to use and terribly error prone, the other is pleasant to use and far less error prone but involves additional expenditure. The cheap method is by using the keyword POKE — we simply POKE every machine code instruction into memory, a byte at a time. Trying to follow how a 'POKEd' machine code program works, or to debug it if it doesn't work, is nothing short of sheer hell. To compare the difference, refer back to the short example which added two numbers together. The left hand version, although still code-like in form, will be shown later to be far more comprehensible than the numeric mess on the right.

By using a piece of software known as an **assembler**, the task is much easier, the machine code listing is easy to read or correct and you are relieved of much boring work. Although this series will cater for readers who are content to POKE everything, all programming examples will be given in assembly language. It cannot be too strongly emphasised that serious machine code work demands the purchase of a good assembler. We have used the MIKRO assembler, which is obtainable as a plug-in cartridge from:
SUPERSOFT,
Winchester House,
Canning Road,
Wealdstone,
Harrow,
Middlesex HA3 7SJ.

Hardware Knowledge Required

If 'hardware knowledge' means familiarity with electronics, then you may rest assured you need hardly any. We shall, from time to time, be mentioning 'volts' but only in yes/no form. That is to say, either a certain voltage is present or is not present. Computers happen to be electronic in nature, but, as far as programming in machine code is concerned, it wouldn't make all that much difference if they ran on North Sea gas. However, you will have to understand a little about the overall system of a computer, particularly the role of the microprocessor, memory chips and a few other bits and pieces — not how they work but what they do. Figure 1 gives a rough idea of how some of the components are connected in most microcomputers. The micrprocessor looks a bit fearsome at this stage but all should become clear as the series progresses.

Many readers will already be aware of the distinction between RAM chips and ROM chips. ROM stands for Read Only Memory and refers to memory chips which hold **permanent** information put there by the manufacturers. Information in ROM is known as **firmware.**

ROMs are said to be non-volatile, meaning that the stored information is still there after the machine is switched off. There are three ROMs in the Commodore 64:

- The 8K operating system ROM known as the **kernal**. This is a program which handles all the mundane operations of the computer such as reading the keyboard, displaying characters on the screen and so on. It occupies the range of hex addresses between E000 and FFFF (59344 to 65535 decimal).
- The 8K BASIC **language interpreter** occupying the hex addresses A000 to BFFF (40960 to 49151 decimal).
- The **character generator** ROM, responsible for arranging the correct pattern of screen dots for every character. There are two separate sets of 256 characters: one set covers the upper case and fixed keyboard graphics, the other set covers the normal upper and lower case typewriter-style characters.

RAM stands for Random Access Memory, an unfortunate title because ROM chips are also 'random access'. The essential feature of RAM chips is the ability to read existing information and write new information under computer control. You may notice from Fig. 1 that the control line, R/W, is connected to the RAM chips but not the ROMs. This is because ROMs are permanently in the 'read' state, whereas RAMs must be switched to a different state when information is to be read from them when they are to be written to. RAMs are more like the old school slate used in the Charles Dickens era — old material can be rubbed off and new material written.

RAMs are **volatile**. Any stored information is lost if the power is interrupted, even for a fraction of a second. RAM is for your use, although some of it is hogged by the operating system for screen display purposes, input/output control and, for want of a better term, working space.

There is indeed 64K of RAM present in the Commodore 64 but, unless some fiddling around is done (explained later in the series) and some ROM facilities sacrificed, only 32K is normally available to the user. It is difficult to proceed any further with computer components or machine code until the binary counting system is explained together with the **hexadecimal** notation.

Unfortunately, the mere mention of binary is enough to invoke a yawn from those

who already know it and a gasp of fear from those who don't.

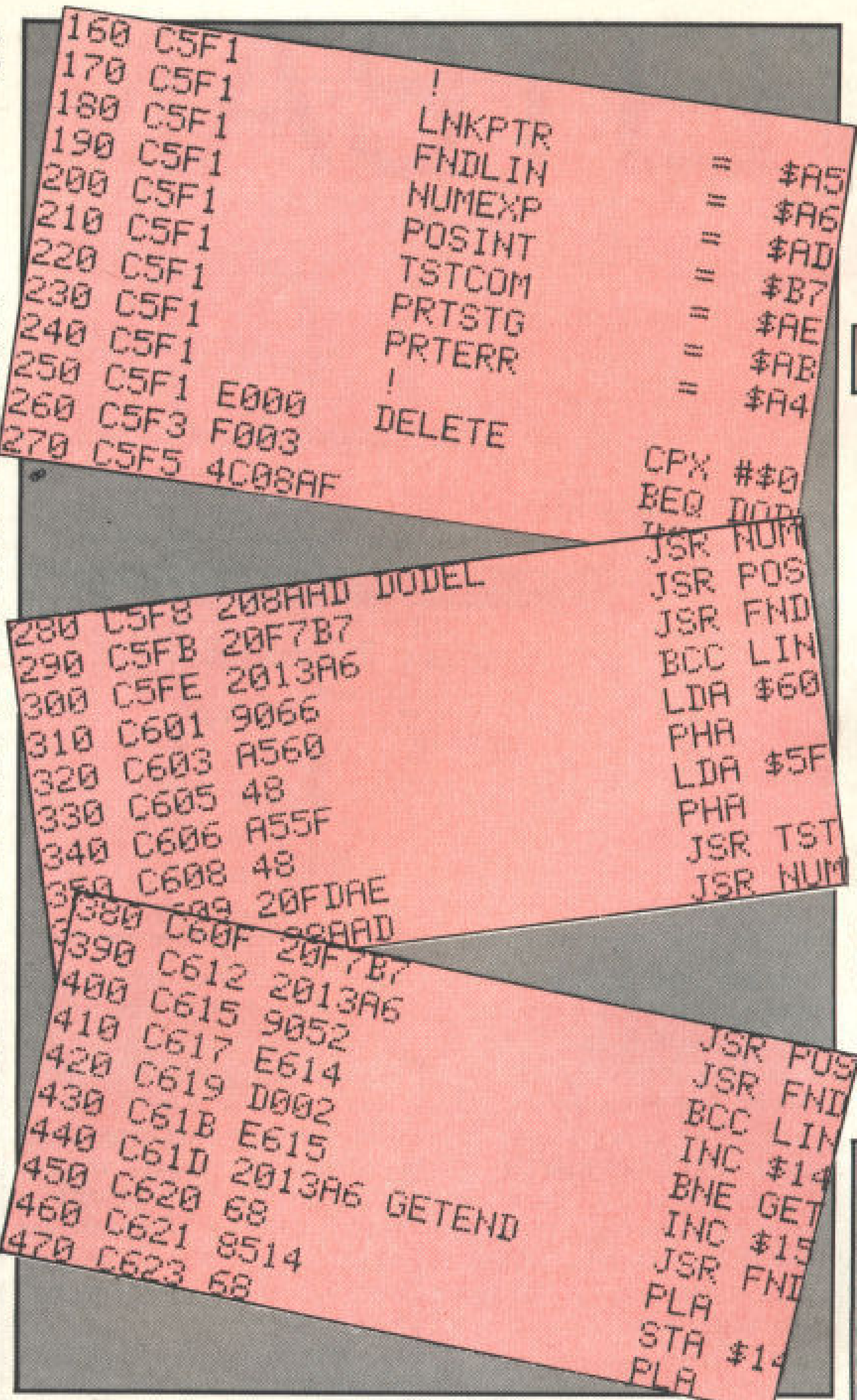
The Binary Number System

There are two kinds of computing machines, the analogue kind (which is of no interest whatsoever to us) and the digital kind. So, when we speak of 'computers', it is taken for granted we are referring to the **digital** kind. A computer is essentially a system of **switches** — silent electronic versions of the ordinary household on/off switch. There are hardly any smoothly varying voltages. Voltages, are either in the HIGH state (about 5V) or the LOW state (nearly zero volts). Due to this essential two-state nature of the computer, it is natural to base all arithmetic and other forms of processing on a counting system which uses only two characters. Binary is such a system because it only uses the characters 1 and 0, allowing us to represent a **HIGH** voltage by 1 and a low voltage by 0.

Any number, however large, can be formed by a string of 1's and 0's called bits. Binary, like the familiar decimal system, uses the normal **place-weighting** system but, instead of each place being worth 10 times the value of the number on its immediate right, it is only worth twice as much. For example 111 in binary is read as one 1, one 2 and one 4 which, in decimal, is 7. To help you get the feel of binary, study the following binary numbers with their decimal equivalents:

1111 = 15
1010 = 10
1110 = 14
0011 = 3
1111 1111 = 255
1000 0000 = 128
1000 0011 = 131

Note that when there are more than four bits, it is conventional to separate them into groups of four because it is easier for humans to read them that way.



Here are a few commonly used terms:

- A **bit** is either a 0 or 1.
- A **nibble** is a group of four bits.
- A **byte** is a group of eight bits.
- The LSB (least significant bit) is the one on the extreme right.
- The MSB (most significant bit) is the one on the extreme left.
- 1K = 1024 (2^{10}, two to the tenth power: nearest power of 2 to 1000).

The following is of great importance:

The number of ways of arranging N bits is 2 ^ N

For example, three bits can only be arranged in 2 ^ 3 = 8 ways as can be seen from the following table:

binary	decimal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Since a byte is the most commonly encountered number of bits, it follows that there are 2 ^ 8 = 256 ways of arranging the bits — there are 256 combinations. Referring back to Fig. 1, you will notice 16 **address** wires coming out of the microprocessor. We can deduce from this that only 2 ^ 16 = 65,536 different address combinations are possible. Now 1K=1024, so there are only 64K valid addresses. This should explain why we cannot have 64K of addressable RAM as well as ROM.

Hexadecimal Notation

The sorts of numbers we can be dealing with in computing get very unweildy when expressed in binary notation; for example, a large number on the address bus could be:

1101 1111 0001 1010

We poor humans need a more compact way of presenting large binary numbers. Well, the four-bit groupings mentioned earlier correspond to a base-16 system of counting (since 2^4 = 16). This counting system is called hexadecimal; if you gain experience with the hexadecimal equivalents, you would be able to say, almost immediately, that the above binary number is DF1A. Just as base two required only two types of digit, 1 and 0, so hexadecimal uses 16 characters, 0 to 9 and the six letters, A to F, to represent a group of four bits. The following table shows the value of each character:

hex	decimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Note that the bit patterns for decimal 0 to 9 are the normal binary equivalents. The last six patterns, which have a decimal value of 10 to 15, are represented by the letters A to F respectively. Once these six extra patterns are mastered (or memorised), it is easy to express a bit pattern, however long and fearsome-looking, in a few hex characters. Remember that each hex character represents one nibble and the following examples should be easy to understand:

1111 0101 1110 0000 1010
F 5 E 0 A
0011 0101
3 5

Changing Hex To Decimal

Each hex digit is worth **sixteen times** as much as the digit on its right. In other words, the system is based on powers of 16 instead of the normal 10 as in decimal. The place weightings are shown below, in exponential as well as decimal form:

16^3	16^2	16^1	16^0
(4096)	(256)	(16)	(1)

Examples based on above weightings:
Hex 111 = 4096+256+16+1 =4369 decimal.
Hex 1032 = 4096+(3x16)+2 =4146 decimal.
Hex 21AF = (2x4096)+256+ (10 x 16)+15=8623 decimal.
Hex FFFF=(15x4096)+(15x256) +(15x16) + 15=65,535 decimal.

After glancing at the above horrible-looking jumble, it may be a relief to learn that only occasionally will it be necessary for you to convert hex to decimal. In fact, you will gradually realise that decimal is alien to machine code. It is actually easier and much quicker, to think in hex, particularly if you do invest in an assembler.

For example, in machine code work, machine addresses are frequently involved. In hex, these addresses within the range 0000 to FFFF hex. In decimal, the same range is 0 to 65,535. Decimal is clumsy and quite unsuited to machine code; hex is concise and tailor-made for it. Memory locations are each one byte 'wide' so the binary contents can be expressed very neatly by two hex characters. The last address in memory is FFFF. There is something final about FFFF but there is nothing final about 65,535!

It does take a little time to get hex into your blood. In the initial stages it comes as a shock to discover that, say 10 in hex is not ten — it is sixteen. Similarly, 32 in hex is not thirty-two — it is fifty. In speech, don't speak of 10 hex as "ten". Call it "one zero" to avoid confusion.

Adding Hex Numbers

Some examples are given below without explanation as an exercise:

13	2F
15	B2
28	E1
FF	CD
01	1A
100	E7

Binary Arithmetic

The pencil and paper procedure for adding binary is the same as for decimal, providing you realise that a carry to the next digit is worth 2 instead of 10. For example:

3	0011
add 5	0101
total	1000

Users of BASIC are usually unaware of the computer's internal arithmetic. Numbers of enormous size are casually entered and the correct answers are taken for granted. When we first take a look into machine code arithmetic, the situation looks decidedly bleak. The 6510 microprocessor is only capable of handling eight bits — in fact, it is called an 'eight-bit' chip. If all eight bits are binary 1's, the largest absolute number it can handle is FF hex or 255 decimal. This may seem a depressing start.

If both positive and negative numbers are to be handled, the situation is even worse because one of the bits is used to indicate the sign of the number. This reduces the maximum positive number to only 127 decimal and the maximum negative number to –128. However, things are not so bad as the appear because, as we shall see later in the series, it is possible to increase the range of numbers by employing some crafty programming tricks.

Two's Complement Notation

Two's complement notation is employed in nearly all computers for the following reasons:

- It enables positive and negative numbers to be handled in an efficient manner
- It simplifies the hardware concerned with addition and subtraction. Only a circuit capable of adding binary numbers is required; subtraction is achieved by adding the two's complement of the number.

The most significant bit in the byte, called the **sign bit**, has the following significance:

The sign bit=0, if the number is positive, and 1 if it is negative.

Examples of positive numbers are given under the following table.

Some examples follow of Two's complement.	
Original	0111 1111
Two's complement	1000 0001
Original	1100 0000
Two's complement	0100 0000
Original	0000 0001
Two's complement	1111 1111

0000 1001 = 09 hex +9 decimal
0111 1111 = 7F hex = +127 decimal.

Negative numbers are not so straightforward because it is not just a question of changing the sign bit. Before we give the rule, try and work it out by studying the following examples:

0000 0001 = +1
0000 1010 = +10
0100 0000 = +64
1111 1111 = –1
1111 0110 = –10
1100 0000 = –64

The term 'flip' in the following rule means change '1's' and '0's' and vice versa: To find the equivalent negative, flip all the bits and then add 1 more. Ignore any final carry. For example:

0000 0011 = +3
Flip all the bits . . . 1111 1100
Add 1. 1
1111 1101 =–3

When we flip all the bits, the result is called the **one's complement** but, adding the final 1, converts it to the **two's complement**. The process works both ways. The rule still applies for changing a negative number back into positive. Try using the rule to change –3 back to +3.

To avoid adding the 1, which may often involve the propagation of a carry, there is an easier way. Assume we start with the positive number:

To obtain the two's complement, start from the right and copy down up to, and including, the first 1. Thereafter, flip the remaining bits.

The first rule we gave is the academic version, the second is the useful one — it is easy to use. Sometimes, we need to find the two's complement of a hex number. The easiest way is to write out the binary bits, use the second rule and put the result back into hex. For example:

3F hex = 0011 1111.
Two's complement = 1100 0001=C1 hex.

This concludes Part 1 of the series and most of the boring background knowledge. We could have skipped a lot of it but this would only have caused trouble later on in the series. A working knowledge of binary and hex is essential for machine code programming.

Get to be a real Commodore and keep the high seas safe with this game from Jamie Clyde.

BATTLE ATAK

BATTLE ATAK IS A computerised version of the popular game 'Battleships'. It runs on the standard Commodore 64 in no more than 10K of memory and uses user-defined graphics.

The scenario is as follows: you are in command of a submarine equipped with torpedoes and a battle computer. Your task is to defend your fleet of surface ships with your torpedoes and destroy the enemy fleet before their torpedoes wipe you out. You are shown your ships on one grid and where your shots hit on the other. Unfortunately you are unable to determine the locations of the enemy vessels, so the second grid starts off blank at the beginning of the game. Both fleets are identical and consist of the following:

1 Aircraft Carrier	(4 squares)
2 Battleships	(3 squares)
3 Submarines	(3 squares)
4 Cruisers	(2 squares)

The figure stated after the ship is the number of squares which it occupies on the grid.

The computer marks up on the grid any developments as the game progresses and the game will finish when either side's fleet has been completely destroyed.



When you run the program

You will first be shown a title page whilst the computer loads in the user-defined graphics into its memory. To proceed, press the Space bar when told and you will be asked for a skill level and then speed level. Level 1 is the easiest game, and beyond level 6 the computer will automatically have another shot if it hits a ship. As far as speed levels are concerned, '1' is the fastest and '5' the slowest — this governs the length of time between each shot.

Once you have entered the desired levels, the grids will be displayed. But before the game can commence, you have to place the ships on your grid which is left of the coordinates in the computer, three characters must be entered. The first is the letter for the distance across, and the others for the figure for the distance down — for example, the coordinate for the top left-hand corner would be 'M01'.

So, to enter a ship on the grid you type the coordinate where you want the bow of the ship to appear, and you then must say if you want the ship to be facing

Explanations of the lines in the program Battle Atak.

Lines	Function
0 - 360	: Title page & Set up graphics
370 - 450	: Set up screen & variables
460 - 630	: Place Player's ships on board
640 - 730	: Set up computer's ships
750 - 1000	: Finalise screen
1000 - 1050	: Main Play routine
1100 - 1600	: Player's shot
2000 - 3000	: Computer's shot
5000 - 6000	: End of game routine
7000 - 8000	: Set Levels
10000 - 11000	: Sound routines
12000 - 20000	: Extra Subroutines
20000 - 30000	: Data for User-Defined Characters & Graphics



up or along. If it cannot fit into your position, an error will appear and you will have to try another point. When all the ships are entered, the game can commence. To play, you enter the coordinates of the point where you want the shot when the top lefthand box says 'Your Turn'. If it is a hit, the part of the ship will be displayed on the right-hand grid. If a computer's shot hits one of your ships, it will turn red.

If a whole fleet is destroyed, the screen clears, and you can have another game if wanted.

Program Listing

```
100 V=53248:S1=54272:POKE808,225
105 POKE53272,21:POKE53280,0:POKE53281,0:PRINT"■"CHR$(8)
110 REM * LOAD CHARACTERS *
115 PRINT"■■■■■■■■■■■■■ CLYDE PRESENTS: "
117 PRINT"■■■■■■■■■■■■■   COMPUTER-"
120 PRINT"■
130 PRINT"■
140 PRINT"■
150 PRINT"■
160 PRINT"■
170 PRINT"■
180 PRINT"◣
190 PRINT"◣
200 PRINT"◣
210 PRINT"◣
215 PRINT"◣
217 PRINT"■
220 PRINT"■
225 PRINT"■
230 PRINT"■
235 PRINT"■
237 PRINT"■
238 PRINT"■
240 A$="     ■  PLEASE WAIT FOR LOADING":A=20:GOSUB19000
300 FORT=0TO215:READA%:POKE14336+T,A%:NEXT
310 FORT=0TO78:READA%:POKE14721+T,A%:NEXT
320 POKE56334,PEEK(56334)AND254:POKE1,PEEK(1)AND251
330 FORT=27*8TO40*8-1:POKE14336+T,PEEK(53248+T):NEXT
340 FORT=58*8TO127*8-1:POKE14336+T,PEEK(53248+T):NEXT
350 POKE1,PEEK(1)OR4:POKE56334,PEEK(56334)OR1
355 FORT=0TO191:READA%:POKE15360+T,A%:NEXT
360 A$="■■       PRESS ANY KEY TO CONTINUE":A=30:GOSUB19000
370 GETR$:IFR$=""THEN370
380 DIMS1(4,4),S2(4,4),S$(4),G(12,12),H(2)
385 GOSUB10500:GOSUB10000
390 FORT=1TO4:FORT1=1TO4:READS1(T,T1):NEXT:FORT1=1TO4:READS2(T,T1):NEXTT1,T
395 FORT=1TO4:READS$(T):NEXT
400 POKE53272,31:SC=1024:CO=55296:P=SC+363:P1=CO+363
407 GOSUB7000:GOSUB10000
410 PRINT"■■■■■■■ BATTLE ATAK BY JAMIE CLYDE■■■"
420 PRINT"■■■■■■■■◣ABCDEFGHIJKLM":FORT=0TO12:PRINT"◣"T;TAB(4)"■■■■■■■■■■■■■■"
425 NEXT
430 PRINT"■■■■■■■■■■"TAB(23)"◣ABCDEFGHIJKLM"
440 FORT=0TO12:PRINT"◣"TAB(19)T;TAB(23)"■■■■■■■■■■■■■■":NEXT
450 PRINT"■■■■■■ YOUR BOARD           MY BOARD"
460 REM * PLACE SHIPS ON GRID *
470 PRINT"■■■■■■PLACE SHIPS:"
480 FORT=1TO4:FORT1=1TO5-T
485 PRINT"■■■■■■■■■■■■■";S$(T)T1;TAB(20)"?";
490 GETR$:IF(R$<"A"ORR$>"M")ANDR$<>CHR$(3)THEN490
492 IFR$=CHR$(3)THENH(2)=27:GOTO5000
495 GOSUB10200
500 PRINT"■"R$;
510 R9=ASC(R$)-65
520 GETR$:IFR$<>"1"ANDR$<>"0"THEN520
525 GOSUB10200
530 PRINTR$;
540 R1=VAL(R$)
550 GETR$:IFR$<"0"ORR$>"9"THEN550
551 GOSUB10200
552 R2=VAL(R$):IFR1=1ANDR2>2THEN550
555 PRINTR$
560 PRINT"■■■■■■■■■■■■■■■■■■■■UP OR ALONG<U/A>"
570 GETR$:IFR$<>"U"ANDR$<>"A"THEN570
575 GOSUB10200:R3=R2:IFR1=1THENR3=R3+10
580 D=1:IFR$="U"THEND=2
590 OND GOSUB18000,18500
600 IFFL=1THENGOSUB16000:GOTO485
610 OND GOSUB17000,17500
620 GOSUB16100:NEXTT1,T
630 :
635 REM * COMPUTER SHIPS *
640 FORT=1TO4:FORT1=1TO5-T
650 X%=RND(1)*12:Y%=RND(1)*12:DI%=RND(1)*2+1:Y=Y%:DI=DI%:X=X%
660 IFX>=12-T ANDDI=1THEN650
670 IFY<=T ANDDI=2THEN650
680 IFDI=2THEN700
690 FORG=1TOT:IFG(X+G,Y)>127THEN650
695 NEXT
697 FORG=1TO4:IFS1(T,G)=0THENNEXT:GOTO750
698 G(X+G,Y)=S1(T,G):NEXT:GOTO750
700 FORG=1TOT:IFG(X,Y-G)>127THEN650
710 NEXT
720 FORG=1TO4:IFS2(T,G)=0THENNEXT:GOTO750
730 G(X,Y-G)=S2(T,G):NEXT
750 NEXTT1,T
752 PRINT"■■■     ┌──────────┐"TAB(21)"┌──────────────┐
754 PRINT"    │          │"TAB(21)"│              │
756 PRINTTAB(4)"└──────────┘"TAB(21)"└──────────────┘
760 PRINT"■■■■■■ ┌──────────────┐ "TAB(19)"┌──────────────┐"
765 PRINT"■ │              │ "TAB(19)"│              │"
770 PRINT"■ └──────────────┘ "TAB(19)"└──────────────┘"
990 GOSUB10000
996 PL=1:H(1)=0:H(2)=0
999 REM * START *
1000 ON PL COSUB 1100,2000
1010 :
1015 FORQ=1TO2:IFH(Q)=27THEN5000
1017 NEXT
1020 GOTO1000
1099 REM * PLAYER *
1100 PRINT"■■■"TAB(5)"■YOUR■  TURN":GOSUB10200:GOSUB10600
1102 PRINTTAB(21)"■■              "
1105 PRINTTAB(22)"TTT■POSITION?     ■■■■";
1110 GETR$:IF(R$<"A"ORR$>"M")ANDR$<>CHR$(3)THEN1110
1115 IFR$=CHR$(3)THENH(2)=27:GOTO5000
1120 R=ASC(R$)-65:PRINTR$;:GOSUB10200
1130 GETR$:IFR$<"0"ORR$>"1"THEN1130
1140 R1=VAL(R$):PRINTR$;:GOSUB10200
1150 GETR$:IFR$<"0"ORR$>"9"THEN1150
1152 GOSUB10200
1155 IFR$>"2"ANDR1=1THEN1150
1157 R2=VAL(R$)
1160 PRINTR$:PRINT"■■■■              "
1170 GOSUB10100:R3=R2:IFR1=1THENR3=R3+10
1175 C1=R3*40+R:IFPEEK(P+C1+20)<>0THENGOSUB1600:GOTO1105
1180 POKEP+C1+20,81:POKEP1+C1+20,7
1190 IFG(R,R3)>127THEN1400
1200 PRINT"■■■■■■"TAB(21)"■   A M I S S  "
1210 PL=2:FORA=1TOSP:NEXT:RETURN
1400 K=G(R,R3):FORT=1TO4:FORT1=1TO4
1410 IFS2(T,T1)=KTHEN1500
1420 IFS1(T,T1)=KTHEN1500
1430 NEXTT1,T:STOP
1500 PRINT"■■■■■■■"TAB(21)"MY ";S$(T)" HIT"
1510 POKEP+C1+20,K:H(1)=H(1)+1
1515 GOSUB10300
1520 FORA=1TOSP:NEXT:PL=2:RETURN
1600 PRINT"■■■■■■■";TAB(21)"■■ FILLED -RETRY "
```



```
1610 RETURN
1998 STOP
1999 REM * COMP. TRY *
2000 PRINT"■■■■■■■■■MY  ":PRINTTAB(22)"]    "
2005 PRINT"■■■■                    "
2007 FORAS=1TOLE+1
2010 X%=RND(1)*13:Y%=RND(1)*13
2020 C1=1+X%+40*Y%:IFPEEK(C1+P1)=4OR(PEEK(C1+P1)=2ANDPEEK(C1+P)>127)THEN2010
2023 IFPEEK(C1+P)>127ANDPEEK(C1+P1)<>2THEN2025
2024 NEXT:IFPEEK(C1+P)>127THENSTOP
2025 GOSUB10400:GOSUB10100
2030 PE=PEEK(C1+P):IFPE>127THEN2400
2040 POKEC1+P,81:POKEC1+P1,4
2050 PRINT"■■■■■■"TAB(3)"  A M I S S  "
2055 FORM=1TOSP:NEXT
2060 PL=1:RETURN
2400 FORT=1TO4:FORT1=1TO4
2410 IFS2(T,T1)=PETHEN2500
2420 IFS1(T,T1)=PETHEN2500
2430 NEXTT1,T:STOP
2500 POKEC1+P1,2:PRINT"■■■■■■"TAB(1)"■A "S$(T)" HIT"
2510 H(2)=H(2)+1:GOSUB10300
2520 PL=1:IFLE>5THENPL=2
2550 FORM=1TOSP:NEXT:RETURN
4998 STOP
4999 REM * END GAME *
5000 KL=1:IFH(2)=27THENKL=2
5010 ONKL GOSUB5500,5700
5020 A$="■■■■■■■■■■WANT ANOTHER GAME?(Y/N)":A=20:GOSUB19000
5030 GETR$:IFR$<>"Y"ANDR$<>"N"THEN5030
5040 IFR$="Y"THEN400
5050 PRINT"■■";:POKE53272,21:POKE53280,14:POKE53281,6:END
5500 PRINT"■■■■■■■■■■YOU ACCOMPLISHED YOUR MISSION"
5510 PRINT"■■■■■■AND HAVE BEEN MADE AN ADMIRAL!"
5520 RETURN
5700 PRINT"■■■■■■■■■■YOU HAVE FAILED YOUR MISSION"
5710 PRINT"■■■■AND THE ENEMY DEFEATED YOUR COUNTRY"
5720 RETURN
6998 STOP
6999 REM * START *
7000 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■BATTLE ZONE"
7010 PRINT"■■■■■■■YOUR MISSION:DESTROY ENEMY FLEET
7020 PRINT"■■■■■■■■■■■ SKILL LEVEL?(0-9)"
7030 GETR$:IFR$<"0"ORR$>"9"THEN7030
7040 GOSUB10200:LE=VAL(R$)
7050 PRINT"■■■■■■■■■■■ SPEED LEVEL?(1-5)"
7060 GETR$:IFR$<"1"ORR$>"5"THEN7060
7070 GOSUB10200:SP=VAL(R$)*500
7080 RETURN
9998 STOP
9999 REM * SOUNDS *
10000 POKES1+4,17:FORC=1TO20:POKES1+1,RND(1)*255:FORC1=1TO10:NEXTC1,C
10010 POKES1+4,16:POKES1+1,0:RETURN
10100 POKES1+4,17:FORC=255TO0STEP-1:POKES1+1,C:NEXT
10110 POKES1+4,16:RETURN
10200 POKES1+11,33:POKES1+8,200:FORC=1TO10:NEXT:POKES1+11,32
10210 RETURN
10300 POKES1+4,129:FORC=50TO-50STEP-1:POKES1+1,ABS(C):NEXT:POKES1+4,128
10310 POKES1+1,0:RETURN
10400 POKES1+11,33:POKES1+8,20:FORC=1TO10:NEXT:POKES1+11,32
10410 RETURN
10500 S1=54272:POKES1+24,15:POKES1+5,255:POKES1+6,1:POKES1+12,25:POKES1+13,1
10510 POKES1+4,17:POKES1+11,17:POKES1+4,16:POKES1+11,16
10520 RETURN
10600 POKES1+11,33:POKES1+8,255:FORC=1TO10:NEXT:POKES1+8,100:POKES1+11,32
10610 RETURN
15998 STOP
15999 REM * MISTAKE *
16000 PRINT"]                    MISTAKE    "
16010 PRINT"T]                        T"
16050 RETURN
16100 PRINT"]                            "
16110 PRINT"T]                        T"
16150 RETURN
16998 STOP
16999 REM * POKE SHIP *
17000 C1=R9+R3*40:FORK=1T04:IFS1(T,K)=0THENNEXTK:RETURN
17005 POKEC1+P1+K,3:POKEC1+P+K,S1(T,K)
17010 NEXT:RETURN
17500 C1=R9+R3*40+1+40:
17505 FORK=1TO4:IFS1(T,K)=0THENNEXTK:RETURN
17510 POKEC1+P1-40*K,3:POKEC1+P-40*K,S2(T,K)
17520 NEXT:RETURN
17998 STOP
17999 REM * CHECK SHIP *
18000 FL=0:IFR9>12-TTHENFL=1
18010 FORJ=1TO4:IFPEEK(P+R9+R3*40+J)>127THENFL=1:RETURN
18020 NEXT
18050 RETURN
18500 FL=0:IFR3>12THENFL=1
18510 IFR3<TTHENFL=1
18520 FORJ=1TO4:I=PEEK(P+R9+R3*40-J*40+1+40):IFI>127THENFL=1:RETURN
18530 NEXT
18550 RETURN
18998 STOP
18999 REM * PRINT *
19000 FORCV=1TOLEN(A$):PRINTMID$(A$,CV,1);:FORZ1=1TOA:NEXTZ1,CV:PRINT
19010 RETURN
19996 STOP
19997 REM ******
19998 REM *DATA*
19999 REM ******
20000 DATA255,129,129,129,129,129,129,255
20005 DATA127,99,99,127,127,99,99,0
20010 DATA126,98,98,126,97,97,127,0
20020 DATA126,96,96,96,96,96,126,0
20030 DATA124,98,98,98,98,98,124,0
20040 DATA126,96,96,120,96,96,126,0
20050 DATA126,96,96,120,96,96,96,0
20060 DATA126,96,96,96,102,98,126,0
20070 DATA98,98,98,126,98,98,98,0
20080 DATA60,24,24,24,24,24,60,0
20090 DATA62,12,12,12,108,108,124,0
20095 DATA99,102,108,124,108,102,99,0
20096 DATA96,96,96,96,96,96,126,0
20097 DATA99,119,107,107,99,99,99,0
20098 DATA115,107,107,107,107,107,103,0
20100 DATA126,98,98,98,98,98,126,0
20110 DATA126,102,102,126,96,96,96,0
20120 DATA126,98,98,98,106,106,126,0
20160 DATA127,99,99,127,108,108,108,0
20170 DATA62,50,48,62,6,54,62,0
20180 DATA126,24,24,24,24,24,24,0
20190 DATA102,102,102,102,102,102,126,0
20194 DATA99,99,99,99,99,54,28,0
20198 DATA97,97,97,109,109,115,97,0
20199 DATA102,102,102,60,60,102,102,0
20200 DATA51,51,51,63,12,12,12,0
20210 DATA126,6,12,24,48,96,126,0
20220 DATA127,99,99,99,99,127,0
20230 DATA24,56,24,24,24,24,60,0
20240 DATA126,6,6,126,96,96,126,0
20250 DATA126,6,6,30,6,6,126,0
20260 DATA96,96,96,108,108,127,12,0
20270 DATA126,96,126,6,6,6,126,0
20280 DATA126,96,126,102,102,102,126,0
20283 DATA126,6,6,6,6,6,6,0
20285 DATA126,102,102,126,102,102,126,0
20290 DATA126,102,102,126,6,6,6,0
20300 REM * GRAPHIC DATA *
20310 DATA0,0,0,255,127,63,31,31
20320 DATA0,0,240,240,255,255,255,255
20330 DATA0,0,0,0,127,255,255,127
20340 DATA16,16,56,56,255,255,255,255
20350 DATA0,0,0,0,252,255,255,254
20360 DATA0,0,8,24,56,120,248,248
20370 DATA248,255,255,255,255,255,248,248
20380 DATA8,24,56,120,248,255,255,255
20390 DATA254,248,255,255,255,255,255,248
20400 DATA255,255,255,248,248,248,255,248
20410 DATA96,96,240,240,240,240,240,240
20420 DATA240,240,240,254,255,255,254,240
20430 DATA240,240,240,240,240,240,96,96
20440 DATA0,0,0,0,255,127,63,31
20450 DATA0,60,60,126,255,255,255,255
20460 DATA0,14,14,206,255,255,255,255
20470 DATA0,0,0,255,255,127,63,31
20480 DATA7,7,7,255,255,255,255,255
20490 DATA224,224,224,255,255,255,255,255
20500 DATA0,0,0,255,255,255,255,255
20510 DATA24,56,120,248,248,248,248,248
20520 DATA248,248,248,248,248,248,255,255
20530 DATA255,255,255,248,248,248,248,248
20540 DATA248,248,248,248,248,248,248,248
20600 REM * DATA FOR SHIPS *
20610 DATA128,129,0,0,134,133,0,0
20620 DATA130,131,132,0,140,139,138,0
20630 DATA141,142,143,0,137,136,135,0
20640 DATA144,145,146,147,151,150,149,148
20700 REM * SHIP STRINGS *
20710 DATA"CRUISER  ","SUBMARINE","BAT SHIP ","A CARRIER"
```

A wondrous utility from Mike Roberts converts machine code into BASIC for those of you with problems.

MACHINE CODE TO BASIC



WHILE I WAS WRITING THE renumber program elsewhere in this issue it became necessary to create a BASIC listing from an area of memory as a list of DATA statements. Also I was doing some work with character generators at the time so I also needed a program that could read the character generator ROM as well (very difficult).

The result is a program that asks you for the start line number of the resulting program and the increment between the lines. Program lines will then begin to flash on the screen. After a while the program will end and the original program will be a lot longer. If you specify a start line number that the program already uses it will overwrite the program and will irrecoverably crash.

The start location for the code to be converted is stored in line 20 (currently 49152) the end location is stored in line 55 (currently 49240-1).

The end result is the original program, which can now be deleted, and a long list of DATA statements containing the code. A POKEr can be put as the first line to POKE in the DATA (like line 10 of renumber) and hey presto there you have it.

The conversion program

```
1 DEF FNH(X)=INT(X/256)
2 DEF FNL(X)=INT(X-(INT(X/256)*256))
3 DEF FND(X)=PEEK(X)+(PEEK(X)*256)
10 INPUT"JQQQQSTART NUMBER, INCREMENT";Z,X
20 POKE56334,PEEK(56334)AND254:POKE1,PEEK(1)AND251:C=49152:REM START LOCATION
30 GOSUB100:PRINT"J";Z;" DATA "+A$:PRINT"GOTO50"
32 POKE832,FNL(Z):POKE833,FNH(Z):POKE834,FNL(X):POKE835,FNH(X)
33 POKE836,FNL(C):POKE837,FNH(C)
35 POKE198,3:POKE631,19:POKE632,13:POKE633,13
40 POKE1,PEEK(1)OR4:POKE56334,PEEK(56334)OR1:STOP
50 POKE56334,PEEK(56334)AND254:POKE1,PEEK(1)AND251
51 DEF FNH(X)=INT(X/256)
52 DEF FNL(X)=INT(X-(INT(X/256)*256))
53 DEF FND(X)=PEEK(X)+(PEEK(X+1)*256)
55 Z=FND(832):X=FND(834):C=FND(836):Z=Z+X:IFC<49240THEN30:REM END LOCATION
60 POKE1,PEEK(1)OR4:POKE56334,PEEK(56334)OR1:END
100 A$=""
110 FORI=0TO9:D=PEEK(C+I):A$=A$+STR$(D)+",":NEXT:A$=LEFT$(A$,LEN(A$)-1)
120 C=C+10:RETURN
```

THE BASIC FACTS PT. 1

Like any language, BASIC can be used well or badly. A P and D J Stephenson brush up your grammar.

BEGINNING A NEW SERIES on BASIC for the 64, or indeed any machine, is a problem. If the standard is pitched too high, new converts are disadvantaged; if too low, the growing mass of knowledgeable readers are inclined to get a little touchy and start muttering things about their intelligence being insulted. Another problem is the Commodore User Manual supplied with the CBM 64. If it was badly written, it would be easy to write an improved version. But it isn't. However, it is a rather thin volume and some users may feel that some of the subjects are given too sparse a treatment for comfort.

It is hoped that this series will help to fill some of the gaps. A certain amount of repetition is inevitable — that is to say, certain descriptions of the BASIC keywords will be little more than a restatement of the Commodore User Manual. However, repetition is not always valueless providing it is not literal repetition. Sometimes, the same facts expressed differently can change darkness into light — or at least twilight. However, it would be pointless to begin this series by defining or describing the purpose of every keyword in the BASIC vocabulary. It is expected that the reader has at least glanced through the Commodore User Manual and that it is kept available as a reference text when following this series.

What Is BASIC?

If we define 'language' as a means of communicating, then BASIC is a primitive language. It is a set of rigidly defined keywords which, by



virtue of a special translation program, can communicate our orders to the computer. The list of **keywords** is the **vocabulary** of the language and the computer will recognise no others. It is impossible to overstress the fact that computers have no initiative whatsoever. If you use a word that is nearly right, the computer's moronic intellect is immediately overstretched so it gives up. The same thing happens if a colon is used when the thing expects a semicolon.

Everything you write must be precise with exact spelling and exact punctuation. The computer doesn't really understand anything — at least not in the normal meaning of the word. It is you, the writer of the program which injects intelligence. Computers themselves may start exhibiting primitive intelligence when the new 'Fifth Generation' machines are launched but, for the moment, we must resign ourselves to the unpalatable fact that all computers, including the CBM 64, have zero intelligence. In spite of this, computers do seem to obey Septimius Sod's fifth law which reads, "If a program has a flaw, a computer will delight in finding it".

The translation program is essential because the computer, or rather the microprocessor chip within the computer, can only respond to a lower level language called 'machine code' (which is covered elsewhere in this issue). As far as BASIC is concerned, it hasn't got a clue. The translation program in the CBM 64 is buried within a ROM (Read Only Memory) chip. Each line in your BASIC program is translated (the technical term is **interpreted**) by the ROM into machine code before it is executed.

This takes place at electronic speed which, in most cases, is so fast that you may be completely unaware of the feverish translation going on behind the scenes. The translation is said to be 'transparent' to the user. And yet it does take time — particularly when your program contains mathematical orgies within loops (loops are bits of program which are executed over and over again until some exit condition is satisfied). Small times (milliseconds) add up to large times (minutes or even hours) if there are enough of them so the translation time can, in some programs, reach an unacceptable limit.

BASIC is not without its critics. In fact it has become fashionable in learned quarters to decry the language. Some of the criticism is justified: BASIC can be slow at times and it is not easy to write programs which obey the rules of good **structure** (structure is a set of programming guidelines designed to make programs easier to read and debug). There is also an element of snobbishness in some of the criticisms. BASIC was designed for ease of learning



and there is little doubt that the original aim has been met. It is the easiest of all computer languages so owners of microcomputers owe a dept to the originators, John Kemeny and Thomas Kurtz of Dartmouth College, USA.

It has been said that the microprocessor brought computers to the high street. This may be true but without BASIC, it is doubtful if the men or women in the high street would have bought many of them. The status languages COBOL, FORTRAN, ALGOL and Pascal have many virtues but simplicity ain't one of 'em. Of course, if you are not interested in programming and treat the computer as a tool to run professional software or to play games, the computer language is of no consequence — but then this series will not interest you anyway.

BASIC Dialects

Natives of London and Glasgow sometimes have communication problems. They both speak the same language but use different dialects. And so it is with BASIC. There are many variations on the original Dartmouth version. The essential features of the original language are preserved but there are some differences. Some dialects

may have additional keywords in the vocabulary or may employ different shades of syntax (grammar). This is understandable. BASIC was first launched in 1964 and the computing scene has changed almost beyond recognition since then. Hardware changes, such as high resolution graphics, joystick controls, graphics 'sprites', sound synthesisers and speech recognition chips, all need appropriate activating keywords.

The more sophisticated the translator, the bigger the ROM. The software in a big ROM costs more and, what is more important, uses up more of the **addressable space** of the computer. For example, the CBM 64 uses a 6510A microprocessor. In common with most other eight-bit microprocessors, only 16 address wires are available, limiting the number of directly addressable memory locations, including both RAM and ROM, to 2∧16 = 64K. Apparently, CBM considered RAM space more important than ROM sophistication so they decided to cut down the BASIC interpreter ROM to 8K. Many would say this is a pity. It is inevitable that penny pinching in this area has led top a rather stingy dialect of BASIC. The best that can be said for it is that it is just adequate.

These gentle (?) criticisms should not be taken to heart. Complex programming projects can be undertaken whatever BASIC dialect is resident in ROM providing we work just a bit harder and be a little more methodical when we write programs.

Programming Guide Lines

If the CBM 64 is your first machine, and you have ironed out the last remaining bug in your first program then, by all means, feel elated. But don't let the elation last too long. The fact that a program works should be considered the first, rather than the last, step. In all probability, the listing will be fouled by scar tissue resulting from multiple alterations and insertions. In all probability, there will be redundant lines and evidence of excessive zeal in the use of GOTOs.



It may be argued that, providing a program works, why bother with niceties? Your first program was probably short and niceities won't have mattered very much. The trouble starts when your confidence increases and you have an urge to tackle more ambitious projects. If you carry on in the original undisciplined manner you will soon regret it. The bug-hunting stage becomes more and more hideous. The process of curing one bug often leads to the introduction of two more. The outcome, after hours or even weeks of work, can end in bouts of verbal obscenities or, even worse, lead to permanent change in the personality.

The answer to all this is to start out on the right lines. Try and make the listing beautiful as well as functional. It is too early in the series to start laying down the precise rules of programming structure but, in the meantime, you would do well to take notice of the guide lines which follow.

Use Of Remarks

Use REMs liberally. It is often thought that remarks are for the benefit of other people trying to decipher how your program works. This is only partially true. They are more of an aid to the programmer. Never rely on memory for understanding the purpose of a routine. It may be perfectly clear at the time of writing, but even a few hours afterwards, you may not be quite so certain how it worked or, in some cases, even what it was supposed to do! REMs use up memory but this doesn't matter in the slightest when you are developing a program. When it is finally finished, the original copy can be kept in its REMmed state and a copy, naked of REMs, kept as a workhorse.

Preliminary diagrams

Scribble out a rough diagram, showing the order in which the various parts of the program are to be carried out. The type of diagram is unimportant to start with. There are things called 'flow charts' and 'top down structure charts' which use established symbolism but there is no urgent need to learn, or even stick to them. Providing you understand your own diagram, that is all that matters. Too much establishment guidance applied too early can crucify enthusiam and destroy initiative.

Variable names

Decide careful on **variable names** and write them down as a list. Unfortunately, CBM BASIC only allows two characters for a variable name, the first of which must be an upper case letter and the second a letter or one of the numbers 0 to 9. Some BASICs allow an unlimited number of characters in a variable name so it is easy to choose them meaningfully. For example, INPUT-VOLTAGE is meaningful but on the 64 we would have to be content with, say, IV instead.

Because of this restriction, it pays to spend some time on the choice of variable names. Although only two characters are allowed, the law of combinations allows you choose any one of 936 combinations. This can easily be demonstrated: the first character must be a letter, so there are 26 ways of choosing one. The second character can be a letter or a numeric figure so there are 36 ways of choosing it. The number of ways in which two characters can be chosen is therefore 26 x 36 = 936 combinations.

Sometimes, it will be necessary to use a name without any meaning (no mnemonic value) in order to avoid clashing with another name. However, whatever names you give them, the most important thing is to write them down first and stick to them like glue. If you neglect this discipline, programming time will squandered by having to continually scroll



back the listing to see if you have used the name before. Giving the same name to (what should be) two different variables can cause distress to the programmer and pleasure to malicious computer.

Modular construction

As far as possible, try and arrange your program as a set of self-contained **subroutines**, each capable of separate testing outside the program. This is sometimes difficult but will pay dividends in the long term. if the subroutine requires some values normally obtained from another part of the program (as it probably will) then write a short trigger program which hands over test numbers, entered from the keyboard. When eventually, you persuade it to pass this test, it can safely be inserted in the program with the knowledge that, whatever else is wrong in the final version that particular subroutine is beyond suspicion. In this way, your program grows as a coherent structure of tested building bricks. (The technical term is **modules**.)

The hated GOTO

Go easy on the GOTOs. The most violent criticism of BASIC has always been directed against the over-use of the GOTO keyword. The GOTO is a jump to a line number: for example, GOTO 560 then, a few lines later, GOTO 1200. Too much of this is called 'spaghetti programming'. It can be absolute hell to follow if carried to excess. GOTOs can never be avoided in 64 BASIC but, with a little care, their population can be decimated.

In any case, don't even think of jumping out of a loop with a GOTO before the normal exit. This is the worst crime in the book, even if the program does work. (There are one or two exceptions to this but this is not the proper time to discuss them.)

Screen messages

However brilliant the program its effect on the user will depend ultimately on the quality of the screen display. There is only one thing worse than a scruffy display and that is one that is over-embellished. Messages to the operator should be as short as possible but not at the expense of clarity. Extremes should be avoided. For example, a message such as "AGE?" is certainly short but it may not be immediately obvious that it is an invitation to input your age at the key board. Staccato messages of this kind often lead to a situation in which screen and user stare at each other blankly, each waiting for the other to do something.

On the other hand, "Would you be kind enough to supply the computer with information as to your age and then press RETURN" is an example of the other extreme and liable to induce projectile vomiting all over the poor keyboard.

The CBM has a liberal assortment of graphic keys and there is a natural temptation in the early stages to tart up every message with a coloured border of circles, squares or asterisks. Over enthusiasm in this respect can reduce the impact of the message by swamping it with the kaleidoscopic beauty (?) of the border.

Programming philosophy

Always program with the idea that the finished product will, in the light of experience, require modifications or additions. This attitude will drive you to plan the program in module form. It will also provide the incentive to include plenty of REMs and, most important of all, it will teach you to cut down on the GOTOs and to keep a register of the variables used. In short, it will help you to write **good** programs — not just ones that work.

Finding And Curing Bugs

To start with bug finding before we have treated programming in detail may seem like putting the cart before the horse but it is based on sound psychological reasoning. Programming is unlike any other discipline because it demands a strange mixture of skills. Attempts to quantify the mixture will inevitably invite argument, but for what it's worth, programming is 70% science, 30% art. Imagination is also an important ingredient providing it is supported by a strong sense of method. But, above all, a programmer must always be on guard against the evils of frustration. It is ridiculously easy to be frustrated, particularly in the early stage of learning. The following adage (which should not be taken too literally) may prove comforting!





1. No program works first time.
2. If it does, it was too simple anyway.

There are two classes of bug:

- Programming errors which are unacceptable to the computer. These are relatively easy to cure because the computer outputs a screen message indicating the nature of the crime. A full list of the error messages are given on pages 151/152 of the Commodore User Manual.
- Programming errors which the computer accepts but cause the program to behave other than the intended behaviour. These bugs are more serious and could take a long time to find.

The most common bug in the first group is due to the use of incorrect grammar. The computer obligingly informs the user than a "SYNTAX ERROR" has been discovered. The offence could be caused by an incorrectly spelt keyword, wrong punctuation, unrecognisable keyword, bracket (parenthesis) missing or one more than there should be, or the incorrect use of arithmetic operators. With regard to brackets, remember there must always be an even number of them. An incorrect arithmetic operator can be more difficult to locate, particularly if the question is at all complex.

The most common error is likely to arise over multiplication. When we write Z = XY in ordinary mathematical notation, it is implied that X and Y are to be multiplied together. The computer does not respond to implications. It wants to know exactly, by using '*' as the multiplication operator, so we must write Z = X*Y.

It should be mentioned here that failure to include the asterix in this example would not necessarily result in a syntax error because Z= XY is recognised as valid syntax. It will be interpreted as a simple assignment, in which the variable named XY is to be assigned to the variable named Z. The example provides a good illustration of the second type of bug. That is to say, the computer accepts it as a valid order but, if the programmer intended multiplication, the result would be quite different to that intended. This kind of error can take some spotting until you get used to arithmetic operators.

While on this subject, it is worth discussing the concept of operator **precedence.** Page 114 of the Commodore User Manual gives the order of precedence in which arithmetical operations are carried out. For example:

V1 = 3+8*5∧2 will make V1 = 203

V1 = 3+8∧2/2 will make V1 = 35

When you are learning, it is safer to forget precedence and rely on parentheses to indicate the order in which the arithmetic is to be applied. The two previous examples would be safer written,

V1 = 3+(8*(5∧2))

V2 = 3+((8∧2)/2

Although parentheses are safer for the inexperienced, they do increase translation time so, in programs where execution time is critical, it is better to revert to precedence.

Syntax errors are responsible for much of the early frustration, mainly because humans are not used to dealing with such fussy animals. It takes some time to come to terms with such unaccustomed intolerence. Just RUN and reRUN the program, pegging away at each offending line, until the program eventually reaches the last line number and then stops. The frustration should then be replaced by a warm glow of pride.

Our intrepid reviewer, Sue Duffield, was very impressed with this artistic program.

ARTISTIC STUFF



WELL, COULD A COMputer teach me what nearly ten years of schooling couldn't, namely to paint? Could it and how! Move over Renoir — here comes Paintpic.

Paintpic is a complete painting and drawing system developed by the New Zealand company Kiwisoft in conjunction with Kuma Computers.

Claimed by the makers to be capable of producing outstanding results, I dutifully worked through the 58-page manual provided to see if I could. Thankfully the manual is very definitely written in layman's terms — it even tells you how to load the cassette which is a boon for the simple-minded. It gently takes the user step by step through the basics of the program, always working by example and you quickly reach the point where you actually start to paint.

The quick load (seven minutes) and the question and answer type entry bring you straight into the drawing mode.

And there you are — Turner's seascapes, Constable's landscapes or even Picasso's madness — all at your fingertips.

The details

So how does it work?

When the program is loaded, a crosshair appears on the TV screen. You begin to paint by moving this across the screen using a joystick or a series of desgnated keys. A trail of minute dots follow whichever direction you choose to go in; left, right, up, down, and even diagonal. Three painting methods are available, pen, brush or text mode.

Using the pen mode there are around 32,000 dots in the painting area from which you make up your pictures. This means your painting can be both intricate and delicate but at the same time it is both painstaking and slow — not recommended for the slapdash among you.

The brush mode means you can colour large areas of the screen quickly and there are individually definable bristles and brushes which you can store and retrieve at will.

The text mode is, as the name suggests, simply the ability to reproduce on the screen any of the keys, letters, figures and symbols, accessed through the keyboard.

You paint in four colours at any given time, chosen from the full colour range of the Commodore 64.

Once you start to paint, if you make a mistake or change your mind, you don't have to wipe the screen clean and start again. Erasing is simply a matter of retracing your steps using a different colour. Who knows how Van Gogh's paintings would have turned out if he'd had this facility?

You can also take all the hard work out of moving the crosshair round and round the screen as you paint because there are nine home positions, accessible by a single key stroke, which take away the necessity of going over and over the same area to get to where you want to be.

A little practice and in no time at all you'll be producing results to stun your friends. And you don't need to lose the masterpiece when you switch the TV off and go to bed. You can save all or part of your painting on tape and retrieve it when you want. The manual also gives detailed instructions on how to photograph your efforts direct from the TV so you can keep them forever.

Very young children can play happily at this game, using the joystick, with no mess or washing up and few tears.

Adults can get equally hooked creating and recreating at little cost to either pocket or temper. Even professional artists, claim the manual, can use the system to produce accurate publishable visual work.

Truly versatile, totally addictive and just fascinating, this is one of the best 'games' I've played for a long time.

Paintpic is for the CBM 64, is priced at £19.95 and comes from Kuma Computers.

Can you outguess the machine? Derek Waldron's programmed a CBM 64 version of this popular game.

CODE BREAKER

THE PROGRAM IS A version of the popular type of code-breaking game. It is a game for two players who take the roles of 'code-setter' and 'codebreaker'. In my version for the Commodore, the computer is given the role of codesetter, leaving you the task of cracking the code.

The code consists of four colours from a choice of eight. Not only do you have to guess which colours have been used, but you also have to determine where each colour has been placed within the code. To help you reach this goal, the computer will mark each guess that you make, according to the following rules: for each colour and position guessed correctly, a black 'peg' will be awarded. Where a colour is guessed correctly, but the position does not match that in the code, a white 'peg' will be awarded. A mark of 4 black 'pegs' therefore means you have cracked the code!

In view of this assistance, you are only allowed 10 attempts in which to break the code. If, after this time, you have been unsuccessful in your attempts, the computer will reveal to you what the code was.

To enter your guess, keys 1-8 should be used, the colour chosen is pressed, a square on the grid will be filled with the appropriate colour, working left to right as each colour is entered. It should be noted that once a colour has been entered it cannot be deleted, so careful thought should be given before entering any colour.

Once the fourth colour has been entered, the computer will automatically



assess your guess and print the appropriate mark in the answer box alongside the line you have entered. The answer will be in the form of black and/or white 'pegs' in accordance with the rules already specified. Please note that the position of any black or white 'pegs' in the answer bears no relation to the position of the colours in your guess. This means that a black 'peg' in the left-most position of the answer does not necessarily mean that the left most colour of your guess is the one that is right.

Watch Out

The relevant points to watch for in each answer given by the computer are the number of 'pegs' and their colours.

If any of your guesses are totally unsuccessful, obviously no pegs will be awarded in the marking, which will be backed up by an unpleasant noise. Do not be disheartened. You will come to learn that an unsuccessful guess will often give your more information than one rewarded with, say, one or two 'pegs'.

The scope for using sound within a program of this nature is limited, and has been restricted to the already mentioned 'rude noise' for an unsuccessful guess, together with different pitched notes for black and white 'pegs', when marked. When you are able to crack the code within the 10 guesses allowed, you will also hear a short, familiar tune.

A summary of the instructions to play the game are included in the program.

The Program

Line 10	Clears the screen and changes the background to grey, with a different shade grey border
Lines 20-90	Prints the first page defining the object of the game
Lines 100-110	Requests a key to be pressed to continue
Lines 112-140	Prints the second page giving instructions to play
Lines 150-160	Requests a key to be pressed to continue
Line 170	Sets up the initial values for the variable array GP(n)
Line 171	Defines A$
Line 172	Defines the variables used for producing sound
Line 180	Sets up the initial values for the variable array AP(n)
Line 190	Sets CT equal to ø
Lines 200-410	Clears the screen and prints the grids on which the guesses and answers are placed
Line 500	Generates a random code of four colours stored in the array C (n)
Line 540	Increments CT by one and checks if 10 guesses have been made
Line 550	Clears G$ and sets up a loop which will be repeated 4 times
Line 560	Requests a key to be pressed
Line 565	Converts the key pressed on an ASCII code, which is assigned to the variable GC
Line 570	Checks that the key pressed is allowable (keys 1-8)
Line 580	Converts the key pressed to a number representing the colour chosen, which is then assigned to the variable G(n)
Line 590	Repeats the above loop until G(n) contains 4 values representing the colours of your guess
Line 600	Clears the flags used in checking your guess
Line 610	Sets up a loop to be repeated four times
Line 620	If guess does not equal code, repeats loop
Line 630	Sets the relevant guess and code flags and increments the black 'peg' counter by one
Line 640	Repeats the loop
Line 650	Sets up two loops used in checking out of position colours
Line 660	Checks if the flags have been set
Line 670	If guess does not equal code, repeats the loop
Line 680	Sets the flags and increments the white peg counter
Line 690	Repeats the loop
Line 700	Goes to the 'print answer' subroutine and then goes back to the request input for next guess
Line 750	Amends variable GP(n) which is used as a POKE address for printing your guess
Line 760	POKEs directly to the screen a 'reversed space'
Line 770	Pokes the relevant colour to the position POKEd with a reverse space in the above line
Line 780	Returns from subroutine
Line 800	Amends the addresses to which the answers are to be POKEd
Line 805	Checks if the answer is all black 'pegs'
Line 810	Checks if the answer is entirely wrong
Line 820	Sets up a loop to be repeated for as many times as there are 'pegs' to be printed
Line 830	POKEs a circle (peg) directly to the screen
Line 840	Checks if any black 'pegs' are to be awarded
Line 850	Decrements the black 'peg' counter by one, POKEs the 'peg' position with the colour black, goes to the 'black peg note' subroutine.
Line 860	GOTO 880
Line 870	POKEs the position of the peg with the colour white and then goes to the 'white peg note' subroutine
Line 880	Repeats the loop
Line 900	Prints 4 black pegs on the answer pad and goes to the 'play a tune' subroutine
Line 920	Asks if you want another go at the bottom of the screen
Line 930	Asks for a key to be pressed
Line 940	Checks if the key was a 'Y' and replays the game if it was
Line 950	If the key pressed was not an 'N' the program goes back to request a key
Line 960	Ends the program
Line 970	Clears the 'another go' question from the bottom of the screen and resets the program to Line 170
Line 1000	Sets volume to maximum. Sets waveform to pulse. Sets high and low pulse rates
Line 1010	Sets high and low frequency values. Sets up sustain value
Line 1020	Play note
Line 1030	Resets high and low frequencies and sustain value
Line 1040	Plays note and then turns voice off
Line 1100	Sets volume to maximum selects sawtooth waveform. Sets attack wave
Line 1110	Sets high and low frequency, and sustain level
Line 1120	Plays note and turns voice off
Line 1200	Sets volume to mid level. Selects sawtooth waveform. Sets attack value
Line 1210	Sets high and low frequency, and sustain level
Line 1220	Plays note and turns voice off
Line 1300	Delay loop
Line 1305	Sets volume to maximum: selects sawtooth waveform. Sets attack level
Line 1310	Reads note values and duration from data statements
Line 1320	Checks if all notes have been read
Line 1330	Sets high and low frequency of note
Line 1340	Sets sustain value
Line 1350	Plays note. Turns voice off. Goes back to play next note
Line 1360	Data containing values of notes to be played in tune.
Line 1500	Updates the position in the array AP(1)-AP(4)
Line 1510	Prints the code if you were unsuccessful in cracking it
Line 1520	Gets the colours of the code
Line 1530	Delay loop before returning to ask if you would like another go.

Variables Used

A$	Set to equal home cursor and 22 cursor down instructions. It is used for printing 'Do you want another go' etc at the bottom of the screen	HF	High frequency	These variables are set to the various appropriate addresses that need to be poked to when using the sound capabilities of the Commodore 64.
		LF	Low frequency	
		PH	High pulse	
		PL	Low pulse	
G$	Is used in the GET G$ statement, when a key input is required	SU	Sustain	
		VO	Volume	
B	Set to equal the number of black 'pegs' to be printed in the answer	WA	Waveform	
		C(n)	Array containing the code to be cracked	
W	Set to equal the number of white 'pegs' to be printed in the answer	G(n)	Array containing your guess	
		AP(n)	Array containing the addresses to which the answer should be poked	
CT	Used to count how many guesses have been made	CF(n)	Array containing the code flags when your guess is marked	
GC	Is set to equall the ASCII code of G$ when when inputting a guess. GC is then to make checked to make sure it is within the range for keys 1-8	GF(n)	Array containing the guess flags when your guess is marked. A flag in GF(n) is set to 2 for a black 'peg', 1 for a white 'peg' or left at ø if that colour does not appear in the code	
AT	Attack			

The addresses 53281, and 53280, used in line 10 control the background and border colours respectively.

The Commodore 64 screen consists of 25 lines of 40 characters each, giving 1,000 possible locations. Each position can be addressed directly by POKEing to the required position on the screen 'memory map'. The map starts at 1024 which corresponds to the top left corner. Hand in glove with the screen map goes the colour map which starts at 55296, also for the top left corner.

Therefore, to put a character directly on screen, two addresses must be POKEd, one for the character, and one for its colour. This concept is used extensively within the program, simplified to POKEing AP(n),X: AP(n)+54272,X(54272) being the difference between 1024 and 55296).

To ensure that the answers and guesses are POKEd to the right positions on the screen, the addresses contained within the arrays AP(n) and GP(n) are incremented by 80, representing two lines down, each time a complete new guess is entered.

The other addresses POKEd to will be found in the subroutines starting at lines 1000, 1100, 1200 and 1300. These are concerned solely with producing sound, and may be left out. If you wish to include sound within the program, you

Program Listing

```
1 REM **********CODE-BREAKER**********
2 REM *****BY D.WALDRON : JAN 1984****
10 PRINT"█":POKE53281,12:POKE53280,15
20 PRINT"████████████***CODEBREAKA***█"
30 PRINT"████YOU HAVE TO CRACK A CODE SELECTED BY   THE COMPUTER."
35 PRINT
40 PRINT"█  THE CODE CONSISTS OF FOUR COLOURS, AND YOUR ATTEMPTS TO CRACK ";
50 PRINT"THE CODE WILL BE MARKED AS FOLLOWS:"
60 PRINT"██  IF YOU GUESS A COLOUR RIGHT, AND IT'S  POSITION IN THE CODE IS THE
";
70 PRINT"SAME, A      '█BLACK█' PEG WILL BE AWARDED."
80 PRINT"█  █IF YOU GUESS A COLOUR RIGHT, BUT IT'S  POSITION IN THE CODE ";
90 PRINT"IS DIFFERENT, A    '█WHITE█' PEG WILL BE AWARDED."
100 PRINT"███         ██PRESS A KEY TO CONTINUE█"
110 GETG$:IFG$=""THEN110
112 PRINT"██████████████***CODEBREAKA***█"
114 PRINT"██  █TO CHOOSE A COLOUR, USE KEYS 1 - 8.    THE COLOUR CHOSEN ";
116 PRINT"WILL BE THE SAME AS   MARKED ON THE FRONT OF THE KEY."
120 PRINT"█████A MAXIMUM OF 10 GUESSES IS ALLOWED TO  CRACK THE CODE, ";
130 PRINT"BUT I'M SURE YOU WON'T  NEED THAT MANY!!!"
140 PRINTTAB(15)"███GOOD LUCK!"
150 PRINTTAB(11)"██████PRESS A KEY TO PLAY█"
160 GETG$:IFG$=""THEN160
170 RESTORE:GP(1)=1113:GP(2)=1116:GP(3)=1119:GP(4)=1122
171 A$="██████████████████████"
172 VO=54296:WA=54276:AT=54277:HF=54273:LF=54272:SU=54278:PH=54275:PL=54274
180 AP(1)=1129:AP(2)=1131:AP(3)=1133:AP(4)=1135
190 CT=0
200 PRINT"█"TAB(10)"██  ***CODEBREAKA***  █"
210 PRINT"███████████┌──┬──┬──┬──┐███┌───────┐"
220 PRINT"███████1██|  |  |  |  |  |       |"
230 PRINT"█████████├──┼──┼──┼──┤   |       |"
240 PRINT"███████2██|  |  |  |  |  |       |"
250 PRINT"█████████├──┼──┼──┼──┤   |       |"
260 PRINT"███████3██|  |  |  |  |  |       |"
270 PRINT"█████████├──┼──┼──┼──┤   |       |"
280 PRINT"███████4██|  |  |  |  |  |       |"
290 PRINT"█████████├──┼──┼──┼──┤   |       |"
300 PRINT"███████5██|  |  |  |  |  |       |"
310 PRINT"█████████├──┼──┼──┼──┤   |       |"
320 PRINT"███████6██|  |  |  |  |  |       |"
330 PRINT"█████████├──┼──┼──┼──┤   |       |"
340 PRINT"███████7██|  |  |  |  |  |       |"
350 PRINT"█████████├──┼──┼──┼──┤   |       |"
360 PRINT"███████8██|  |  |  |  |  |       |"
370 PRINT"█████████├──┼──┼──┼──┤   |       |"
380 PRINT"███████9██|  |  |  |  |  |       |"
390 PRINT"█████████├──┼──┼──┼──┤   |       |"
400 PRINT"███████10██|  |  |  |  |  |      |"
410 PRINT"█████████└──┴──┴──┴──┘   └───────┘"
500 FORX=1TO4:C(X)=INT(8*RND(1)):NEXT
540 CT=CT+1:IFCT=11THEN1500
550 G$="":FORX=1TO4
560 GETG$:IFG$=""THEN560
570 GC=ASC(G$):IFGC<49ORGC>56THEN560
580 G(X)=(VAL(G$))-1:GOSUB750
```

CODE BREAKER

should be able to insert commands recognised by your own machine quite easily.

Deleting colours

Should anyone feel that it would be desirable to have the ability to delete a colour before all four have been entered and marked, the following may be of use to you.

Amend line 565 to read: 565 GC=ASC(G$):IF GC=20 THEN 1600

Insert these lines: 1600 IF X=1 THEN 550

1610 X=X-1: POKE GP(X)+54272,12:POKE GP(X)+54273, 12

1620 GP(X)=GP(X)—80: GOTO 560

This will only work on the line you are currently entering, providing the fourth colour has not been entered.

Hints

The program should be easy to convert to other home micros. Reversed characters within the quotes will be one of the following:

Reversed Heart	Clears the screen and places the cursor in the home position (top left of the screen)
Reversed S	Places the cursor in the home position without clearing the screen
Reversed R	Turns the reverse video on
Reversed —	Turns reverse video off
Reversed Q	Cursor down one position
Reversed]	Cursor right one position

```
590 NEXT
600 FORX=1TO4:GF(X)=0:CF(X)=0:NEXT
605 B=0:W=0
610 FORX=1TO4
620 IFG(X)<>C(X)THEN640
630 GF(X)=2:CF(X)=1:B=B+1
640 NEXT
650 FORY=1TO4:FORX=1TO4
660 IFGF(X)=1ORGF(X)=2ORCF(Y)=1THEN690
670 IFG(X)<>C(Y)THEN690
680 GF(X)=1:CF(Y)=1:W=W+1
690 NEXTX,Y
700 GOSUB800:GOTO540
750 GP(X)=GP(X)+80
760 POKEGP(X),160:POKEGP(X)+1,160
770 POKEGP(X)+54272,G(X):POKEGP(X)+54273,G(X)
780 RETURN
800 FORX=1TO4:AP(X)=AP(X)+80:NEXT
805 IFB=4THEN900
810 IFB=0ANDW=0THENGOSUB1000:RETURN
820 FORX=1TO(B+W)
830 POKEAP(X),81
840 IFB=0THEN870
850 B=B-1:POKEAP(X)+54272,0:GOSUB1100
860 GOTO880
870 POKEAP(X)+54272,1:GOSUB1200
880 NEXT:RETURN
900 FORX=1TO4:POKEAP(X),81:POKEAP(X)+54272,0:GOSUB1100:NEXT:GOSUB1300
920 PRINTA$TAB(7)"██ANOTHER GO (Y/N)█";
930 GETG$:IFG$=""THEN930
940 IFG$="Y"THEN970
950 IFG$<>"N"THEN930
960 END
970 PRINTA$TAB(10)"                    ";:GOTO170
1000 POKEVO,15:POKEWA,65:POKEAT,190:POKEPH,8:POKEPL,0
1010 POKEHF,1:POKELF,205:FORT=1TO30:POKESU,136:NEXT
1020 FORT=1TO100:NEXT:POKEHF,0:POKELF,0
1030 POKEHF,1:POKELF,105:FORT=1TO30:POKESU,136:NEXT
1040 FORT=1TO300:NEXT:POKEHF,0:POKELF,0:POKEWA,0:RETURN
1100 POKEVO,15:POKEWA,33:POKEAT,16
1110 POKEHF,96:POKELF,254:FORT=1TO20:POKESU,136:NEXT
1120 FORT=1TO150:NEXT:POKEHF,0:POKELF,0:POKEWA,0:RETURN
1200 POKEVO,10:POKEWA,33:POKEAT,16
1210 POKEHF,24:POKELF,63:FORT=1TO20:POKESU,136:NEXT
1220 FORT=1TO150:NEXT:POKEHF,0:POKELF,0:POKEWA,0:RETURN
1300 FORT=1TO500:NEXT
1305 POKEVO,15:POKEWA,33:POKEAT,36
1310 READH,L,D
1320 IFH=-1THENRETURN
1330 POKEHF,H:POKELF,L
1340 FORT=1TO30:POKESU,132:NEXT
1350 FORT=1TOD:NEXT:POKEHF,0:POKELF,0:POKEWA,0:GOTO1305
1360 DATA28,214,250,32,94,250,25,177,350,6,108,200,9,159,400,-1,-1,-1
1500 FORX=1TO4:AP(X)=AP(X)+80:NEXT
1510 PRINTA$TAB(9)"█THE CODE WAS";:FORX=1TO4:POKEAP(X),81
1520 POKEAP(X)+54272,C(X):NEXT
1530 FORX=1TO5000:NEXT:GOTO920
```

RENUMBERING

Mike Roberts has come to the rescue of tangled line numbers with this great renumbering utility for the CBM 64.

COMMODORE BASIC IS lacking in a great deal of functions. Graphics commands have been well documented and most magazines have published programs that do hi-res, sprites, sound etc.

Utilities tend to be a bit thin on the ground as everybody's efforts are directed against the graphics. While using Supersoft's 'Mikro' assembler I suddenly needed a renumber program. The following machine code program was the result. As I only use it for renumbering assembler programs I'm afraid that there is no facility for renumbering GOTOs GOSUBs etc as this would make the program quite long and complex.

As it stands it is just as good as the renumber program in Commodore's Simon's BASIC cartridge.

Setting Great Store

To explain how the program works it is first necessary to go into detail about how BASIC stores a program line. The first byte in a program line is zero (not the last byte as most people seem to think), the next two bytes contain the address of the start of the next program line. This is called the link address and is used by the interpreter only while running a program and inserting a new line. The next pair of bytes is the pair that we're interested in — the line number. After the line number is the rest of the BASIC line.

The line number is stored as a 16 bit binary unsigned integer, so it can be from 0 to 65535. So for a renumber program all we have to do is scan through the program picking out the zero bytes, indicating the start of a line, skip a few and modify the two bytes containing the line number — easy!

Of course there are some options. You can dictate the line number that the program starts with and the difference between them. This is done by POKEing four bytes into the 64's memory:

POKE 248, INT(X/256): POKE247,X–(INT(X/256)★256)

This will set the start number for the program where 'X' should be replaced by the number that you want the program to start with.

POKE250,INT(Y/256): POKE 249,Y–(INT(Y/256)★256)

This will set set the difference between the lines, where 'Y' should be replaced by the increment.

The machine code should then be called with 'SYS 49152', after a short pause the program will be renumbered. The POKED values will be preserved, so you only have to POKE them the first time, the next time they will still be there and you just need to do the SYS call.

Machine Code Program

```
10 *=$C000
20 START=$F7
30 INCR=$F9
40 PTR=$FB
50 COUNT=$FD
60 LDX START
70 STX COUNT
80 LDX START+1
90 STX COUNT+1
100 LDX #$08
110 STX PTR+1
120 LDX #$03
130 STX PTR
140 LDY #$00
150 NEXTLN LDA COUNT
160 STA (PTR),Y
170 JSR INPT
180 LDA COUNT+1
190 STA (PTR),Y
200 CLC
210 LDA COUNT
220 ADC INCR
230 STA COUNT
240 LDA COUNT+1
250 ADC INCR+1
260 STA COUNT+1
270 LOOP JSR INPT
280 LDA (PTR),Y
290 CMP #$00
300 BNE LOOP
310 INY
320 LDA (PTR),Y
330 CMP #$00
340 BNE NXT
350 INY
360 LDA (PTR),Y
370 CMP #$00
380 BEQ ENDIT
390 NXT LDY #$00
400 JSR INPT
410 JSR INPT
420 JSR INPT
430 JMP NEXTLN
440 ENDIT RTS
450 INPT INC PTR
460 BNE NEXTP
470 INC PTR+1
480 NEXTP RTS
```

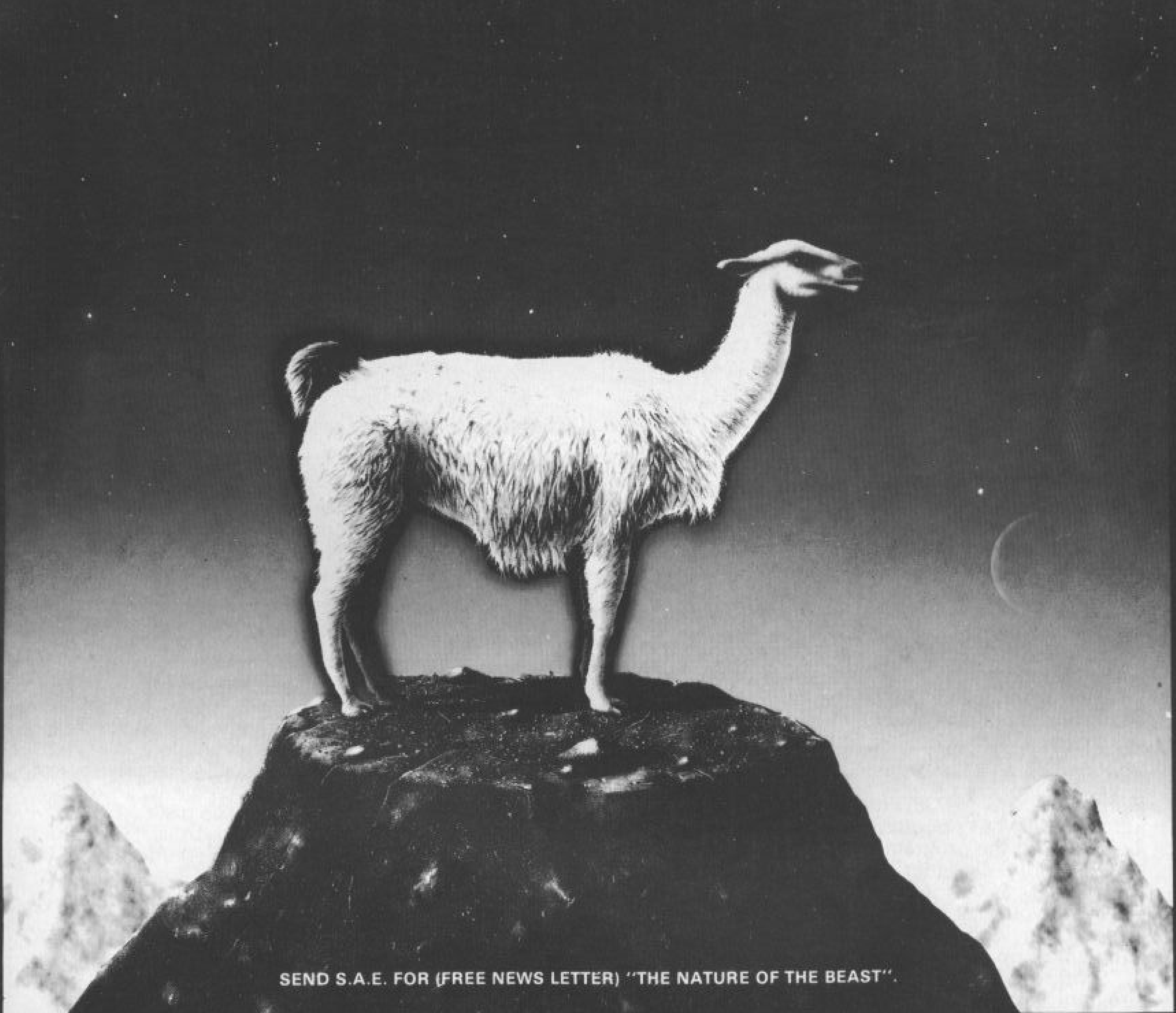
BASIC Loader

```
1 REM RENUMBER BY MIKE ROBERTS JULY 1984
10 FORI=49152TO49240:READJ:POKEI,J:NEXT
20 DATA  166, 247, 134, 253, 166, 248, 134, 254, 162, 8
30 DATA  134, 252, 162, 3, 134, 251, 160, 0, 165, 253
40 DATA  145, 251, 32, 80, 192, 165, 254, 145, 251, 24
50 DATA  165, 253, 101, 249, 133, 253, 165, 254, 101, 250
60 DATA  133, 254, 32, 80, 192, 177, 251, 201, 0, 208
70 DATA  247, 200, 177, 251, 201, 0, 208, 7, 200, 177
80 DATA  251, 201, 0, 240, 14, 160, 0, 32, 80, 192
90 DATA  32, 80, 192, 32, 80, 192, 76, 18, 192, 96
100 DATA  230, 251, 208, 2, 230, 252, 96, 255, 255, 255
```

Llamasoft

ORIGINAL SOFTWARE DESIGN

49 MOUNT PLEASANT, TADLEY, HANTS. RG26 6BN.



SEND S.A.E. FOR (FREE NEWS LETTER) "THE NATURE OF THE BEAST".



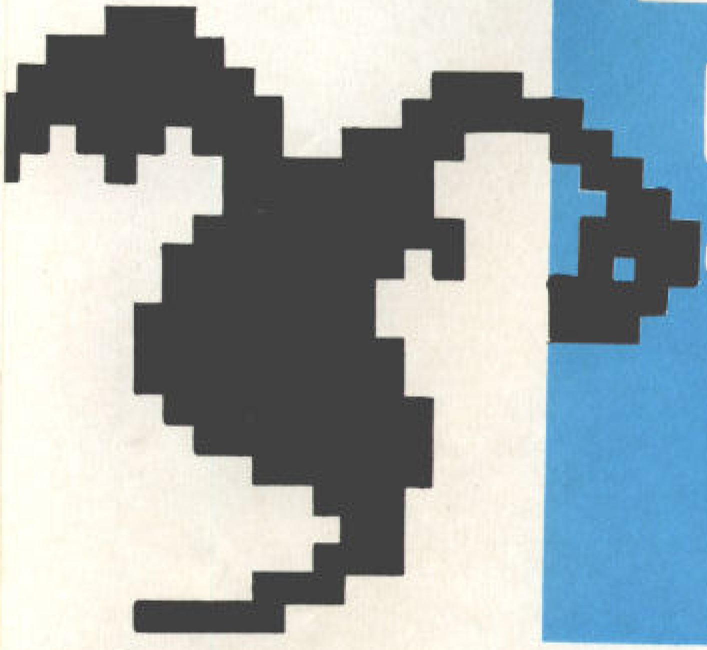
Game	Formats
TRAXX	VIC 20 LLA 12003 £5.00
HELL GATE	VIC 20 LLA 12007 £6.00 CBM 64 LLA 21008 £5.00
METAGALACTIC LLAMAS BATTLE	VIC 20 LLA 11006 £5.50 CBM 64 LLA 21010 £5.50
ABDUCTOR	VIC 20 LLA 11001 £5.00
MATRIX	VIC 20 LLA 12005 £5.00 CBM 64 LLA 21006 £6.00
LASERZONE	VIC 20 LLA 12004 £5.00 CBM 64 LLA 21004 £6.00
GRIDRUNNER	VIC 20 LLA 11002 £5.00 CBM 64 LLA 21003 £5.00 ATARI LLA 41001 £5.00
SHEEP IN SPACE	CBM 64 LLA 21009 £7.50
HOVER BOVVER	CBM 64 LLA 21005 £7.50
ATTACK OF THE MUTANT CAMELS	CBM 64 LLA 21001 £6.00 ATARI LLA 41003 £7.50
REVENGE OF MUTANT CAMELS	CBM 64 LLA 21007 £7.50

GSH · WOOLWORTH · W H SMITH · Boots · John Menzies · Access

DEALERS may order direct from CBS Telesales 01-960 2155 quoting account number, LLA code number and quantities. Goods delivered within 48 hours.

Painting in the pixels was never easier! Design your own sprites with this package from J McHale and A Carton.

SPRITE DESIGNER '64



TYPE IN THE PROGRAM provided and save it on a blank cassette. Now RUN it and if all has gone as planned, a message will appear on the screen, telling you to place a blank cassette in your C2N unit so that 'Sprite Designer '64' may be SAVED to tape as a machine code file.

After it has been SAVED, verify it to ensure that there are no errors present.

To load 'Sprite Designer '64', type 'LOAD"", 1,1 — then type 'SYS (64738)' to reset the CBM 64's pointers before using the program.

As 'Sprite Designer' is located at $C000 (49152 decimal), you may use it in conjunction with a BASIC program of up to 12K.

Sprites designed by this program are located between pages 230 and 254 inclusive ie (230 x 64) — ((253 x 64) + 63).

You may design a total of 25 sprites which should be sufficient.

SYS (50000) initialises the sprite editor.

Movement is by means of the cursor keys, which I found to be the most practical, with the Shift key playing its usual role ie CRSR = Right: CRSR + SHIFT = Left:

When initialised, 'Sprite Designer '64' displays a grid and four sprites on the right hand side of the screen, with the markings 's', 'x', 'y' & 'x&y' displayed alongside.

These letters stand for: standard sprite, X — expanded, Y — expanded and X&Y expanded respectively.

Current sprite status & Page No. is displayed at the top of the screen and you should see a cursor flashing in the top left hand corner of the 24 x 21 grid.

Every bit set in the sprite block will be displayed on the grid as a '•' and every blank bit, a '+'.

Remember to be careful when using the 'clear grid' function' as it also wipes the sprite definition.

It is a good idea to familiarise yourself with the keys before starting to define sprites.

RUN/STOP — Restore has been disabled for programming purposes but is re-enabled on exiting to BASIC.

Useful Locations

($C8A7) — 51367 is the only location to use to the user. It is used in the timing of keyboard response etc.

The value of this location is set at 40.

Changing this value will make key response faster or slower, 1 = fastest; 255 = slowest.

Explanation

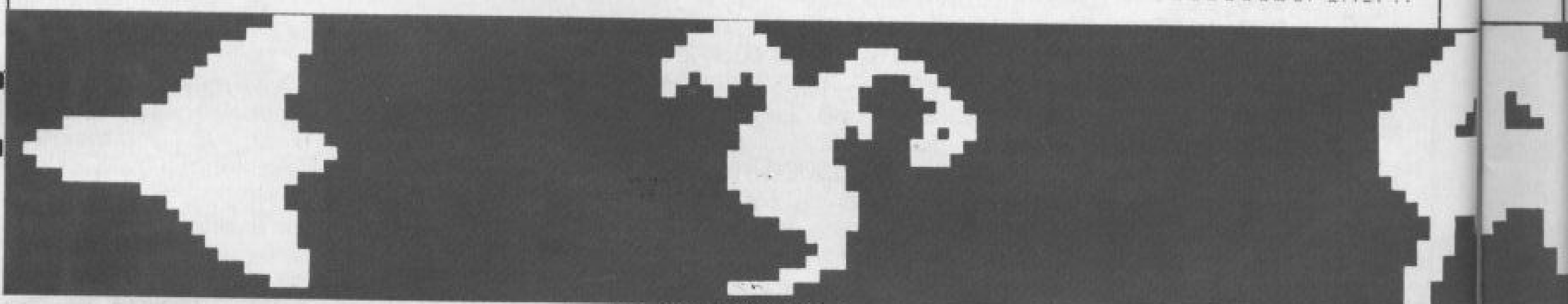
Function	Description	Key Used:
Clear Grid.	Clears sprite grid and current sprite block.	CLR/HOME
Page Plus.	Advances to next sprite page.	'+'
Page Minus.	Lowers sprite page.	'–'
Exit.	Exit back to BASIC	'X'
Save.	Save sprite definitions to tape.	'S'
Enable Multicolour.	Enable sprite Multicolour mode.	'F1'
Disable Multicolour.	Disable sprite Multi-colour mode.	'F1'
Change Sprite colour		'F3'
Change M-col Reg 1.		'F5'
Change M-col Reg 2.		'F7'
Rotate Sprite.	Rotates sprite through 180° (Horizontally)	'R'
Invert Sprite.	Rotates sprite though 180° (Vertically)	'I'
Fill.	Fills in one bit of the sprite definition.	'SPC'
Delate.	Rubs out one bit of the sprite definition.	'SPC'

Program Listing

```
10 REM ***********************
15 REM *                     *
20 REM * SPRITE DESIGNER '64 *
25 REM *                     *
30 REM *         BY          *
35 REM *                     *
40 REM * J.MC HALE & A.CARTON*
45 REM *                     *
50 REM * SLIGO,EIRE ........ *
55 REM *                     *
60 REM ***********************
100 POKE53281,9:POKE53280,9
110 PRINT"ϽπSPRITE DESIGNER '64 WRITTEN BY J.MC HALE"
120 PRINTTAB(25);"≡ANDπ";TAB(31);"A.CARTON"
130 GOSUB800:GOSUB900
140 PRINT"QQQQQQQQQ≡PLEASE WAIT - LOADING CODE INTO MEMORY !"
150 SYS53000
160 SA=49152:TL=0
170 READHX$:IFHX$="XX"THEN300
180 FORC=0TO7
190 H=ASC(MID$(HX$,C*2+1,1))-48:IFH>9THENH=H-7
200 L=ASC(MID$(HX$,C*2+2,1))-48:IFL>9THENL=L-7
210 MC=H*16+L:POKESA+C,MC:TL=TL+MC:NEXTC:SA=SA+8:GOTO170
300 POKE53002,49:POKE53007,234:SYS53000
310 PRINT"Ͻ";
320 IFSA<>51680ORTL<>353013THEN500:REM * TRAP ERRORS *
330 REM * EVERYTHING ALRIGHT *
335 POKE53280,14:POKE53281,6
340 PRINT"Q≡THE CODE HAS BEEN LOADED INTO MEMORY    Ͻ"
350 PRINT"PLACE BLANK CASSETTE INTO YOUR C2N UNIT."
360 PRINT"WHEN READY,PRESS '≡*Ͻ' & '≡SPRITE DESIGN64Ͻ'"
370 PRINT"WILL BE SAVED TO TAPE AS A MACHINE CODE "
380 PRINT"FILE ≡..................................Ͻ"
390 KP=PEEK(197)
400 IFKP<>49THEN390
410 SYS52016:FORTM=0TO2000:NEXT
420 PRINT"ϽQ≡'SPRITE DESIGN64' HAS BEEN SAVED ON TAPE";
430 PRINT"IN MACHINE CODE FORMAT.TO RE-LOAD, TYPE Ͻ"
440 PRINT"LOAD";CHR$(34)""CHR$(34);",1,1"
445 PRINT"QQQQQQQQQ";TAB(13);"Q≡VERY IMPORTANT"
450 PRINT"QQϽVERIFY 'SPRITE DESIGN 64' TO ENSURE THAT"
460 PRINT"IT HAS BEEN SAVED WITHOUT ERRORS ......."
470 POKE53002,21:POKE53007,207:SYS53000
480 END
500 REM * ERROR TRAPPING *
510 PRINT"Ͻ≡Ͻ";
520 IFSA<51680THENPRINT"INSUFFICIENT DATA ERROR ?":GOTO535
530 IFSA>51680THENPRINT"EXCESSIVE DATA ERROR ?"
535 PRINT"CHECK DATA LIST CAREFULLY"
540 IFTL=353013THEN580
550 PRINT"QQCHECKSUM ERROR ? - CHECK DATA CAREFULLY"
560 PRINT"FOR INCORRECT ENTRIES.................."
580 END
800 REM * SAVE TO TAPE ROUTINE *
810 FORT=0TO49:READA:POKE52000+T,A:NEXTT:RETURN
820 DATA83,80,82,73,84,69,32,68,69,83,73,71,78,32,54,52
830 DATA169,0,133,248,169,192,133,249,169,1,168,170
840 DATA32,186,255,169,16,162,32,160,203,32,189,255
850 DATA169,248,162,224,160,201,32,216,255,96
900 REM * INTERRUPT ROUTINE     *
910 FORT=0TO39:READA:POKE53000+T,A:NEXTT:RETURN
920 DATA120,169,21,141,20,3,169,207,141,21,3,88,96,198,2
930 DATA240,3,76,49,234,169,8,133,2,162,40
940 DATA189,223,5,73,128,157,223,5
950 DATA202,208,245,76,49,234
```

Program Listing

```
1000 DATAA99320D2FFA900AA
1005 DATAA88D20D08D21D0EA
1010 DATAEAEAEAEAEAEAEAEA
1015 DATAEAEAEAA97B85F885
1020 DATAFAA90485F9A9D885
1025 DATAFBA95B91F8A90B91
1030 DATAFAA5F818692885F8
1035 DATA85FAA5F9690085F9
1040 DATA69D485FBE8E015D0
1045 DATAE0A200C8C018D0CB
1050 DATAEAA93785F8A200A5
1055 DATAF89D53049D5B049D
1060 DATA6304A9039D53D89D
1065 DATA5BD89D63D8C6F8E8
1070 DATAE008D0E3A95085F8
1075 DATA85FAA90485F9A9D8
1080 DATA85FBA200A000A9A0
1085 DATA91F8A90691FAA01D
1090 DATAA9A091F8A90691FA
1095 DATAA5F818692885F885
1100 DATAFAA5F9690085F969
1105 DATAD485FBE8E016D0D4
1110 DATAA97885F885FAA904
1115 DATA85F9A9D885FBA930
1120 DATA85FC85FDA200A001
1125 DATAA5FC91F8A90891FA
1130 DATAC8A5FD91F8A90891
1135 DATAFAA01BA5FC91F8A9
1140 DATA0891FAC8A5FD91F8
1145 DATAA90891FAA5F81869
1150 DATA2885F885FAA5F969
1155 DATA0085F969D485FBE6
1160 DATAFDA5FDC93AD006A9
1165 DATA3085FDE6FCE8E015
1170 DATAD0B4A200A9A09DC0
1175 DATA079D2804A9069D28
'180 DATAD89DC0DBE8E01ED0
1185 DATAEBA9138D4704A918
1190 DATA8DE7048D9F06A919
1195 DATA8D87058DA106A926
1200 DATA8DA006A9018D47D8
1205 DATA8DE7D88D87D98D9F
1210 DATADA8DA0DA8DA1DAA2
1215 DATA1EBD67C19DFF03A9
1220 DATA019DFFD7CAD0F260
1225 DATA939092899485A084
1230 DATA859389878EA7B6B4
1235 DATAA08299A08AAE8D83
1240 DATAA088818C85A0A900
1245 DATA85FBA2080A26FB06
1250 DATAF9900718 65F89002
1255 DATAE6FBCAD0EF85FAEA
1260 DATAA5FA8DABC1A5FB8D =
1265 DATAACC1608039EAEAEA
1270 DATAA240A9009DBF3FCA
1275 DATAD0FAA9E18D2803EA
1280 DATAEAEAEAEAEAEAEAEA
1285 DATAEAEAEAEA60A210A9
1290 DATA009DFFCFCAD0FA8D
1295 DATA15D08D10D060A9FF
1300 DATA8DFF07A207A9E69D
1305 DATAF707CAD0FA602042
1310 DATA2062208220BEA208
1315 DATABDEDC19DFFCFCAD0
1320 DATAF7A90F8D15D08D10
1325 DATAD0A90C8D17D0A90A
1330 DATA8D1DD0A208A9019D
1335 DATA26D0CAD0FA60A200
1340 DATAA9FF9DC03FE8E8E8
1345 DATAE018D0F660A9E68D
1350 DATAFFCFA9308DFECFA2
1355 DATA328EFCCFE88EFDCF
1360 DATA60AD10D0290F8D10
1365 DATAD0A9308D0ED0A94A
1370 DATA8D0FD0AD15D00980
1375 DATA8D15D0A90385FCA9
1380 DATA7885FDA90485FEA9
1385 DATA808DF1CF60A90085
1390 DATAF885FAA90485F9A9
1395 DATAD885FBA200A000B1
1400 DATAF8C951D004A90191
1405 DATAFAC8C028D0F1A5F8
1410 DATA18692885F885FAA5
1415 DATAF9690085F969D485
1420 DATAFBE8E019D0D760A9
1425 DATA0385FCA97885FDA9
1430 DATA0485FEADFFCF85F8
1435 DATAEAA94085F92086C1
1440 DATAADABC185B9ADACC1
1445 DATA85BAA000A208B1B9
1450 DATA0A900B48A95185BB
1455 DATA20F0C24CE6C248A9
1460 DATA5B85BB20F0C268CA
1465 DATAD0E6C8C03FD0DD20
1470 DATA6DC26098488A48A4
1475 DATAFCA5BB91FDC8C01B
1480 DATAD00FA5FD18692885
1485 DATAFDA5FE690085FEA0
1490 DATA0384FC68AA68A860
1495 DATAEAEAC5C5F0FC8DF6
1500 DATACF60AD1CD0490F8D
1505 DATA1CD060A200ADFECF
1510 DATA9D27D0E8E004D0F8
1515 DATAEEFECF60EE25D060
1520 DATAEE26D060EAEAEAEA
1525 DATAEAEAEAEAEAEAEAEA :
1530 DATA688DFBCF688DFACF
1535 DATAA9E68DFFCFA9308D
1540 DATAF9CFA2328EF7CFE8
1545 DATA8EF8CF20B0C120CD
1550 DATAC120DEC1201EC2A9
1555 DATA008D1CD0A90A8D25
1560 DATAD0A9068D26D0A993
1565 DATA20D2FFA9008D20D0
1570 DATA8D21D02000C020A6
1575 DATAC78DEDCFEA4CF9C3
1580 DATAADFFCF186901D001
1585 DATA608DFFCFADF9CF18
1590 DATA6901C93AD009A930
1595 DATA8DF9CFEEF8CF608D
1600 DATAF9CF60ADFFCF38E9
1605 DATA01C9E5D001608DFF
1610 DATACFADF9CF38E901C9
1615 DATA2FD009A9398DF9CF
1620 DATACEF8CF608DF9CF60
1625 DATAADFACF48ADFBCF48
1630 DATA6053505249544553
1635 DATA2EA98E20D2FFA908
1640 DATA20D2FF4CB7C7D2FF
1645 DATAA90820D2FFA98E20
1650 DATAD2FFA9084CB7C700
1655 DATA00ADFFCF85F8A940
1660 DATA85F92086C1ADABC1
1665 DATA85F8ADACC185F9A0
1670 DATA00A9008502AD1CD0
1675 DATAD00620AEC44C43C4
1680 DATA2080C4A50291F8C8
1685 DATAC040D0E560ADFFCF
1690 DATA85F8A94085F92086
1695 DATAC1A200A000B1FA48
1700 DATAA002B1FAA00091FA
1705 DATAA0026891FAA5FA18
1710 DATA690385FAA5FB6900
1715 DATA85FBE8E015D0DC60
1720 DATAB1F8480A0A0A0A0A
1725 DATA0A850268484A4A4A
1730 DATA4A4A4A1865028502
1735 DATA684829304A4A1865
1740 DATA028502682 90C0A0A
1745 DATA1865028502 60B1F8
1750 DATAA2084AB007480602
1755 DATA684CC6C4480602A5
1760 DATA0209018502 68CAD0
1765 DATAE960ADFFCF85F8A9
1770 DATA4085F92086C1A5FA
1775 DATA18693C85F8A5FB69
1780 DATA0085F9A200A000B1
1785 DATAFA48B1F891FA6891 :
1790 DATAF8C8C003D0F1A5FA
```



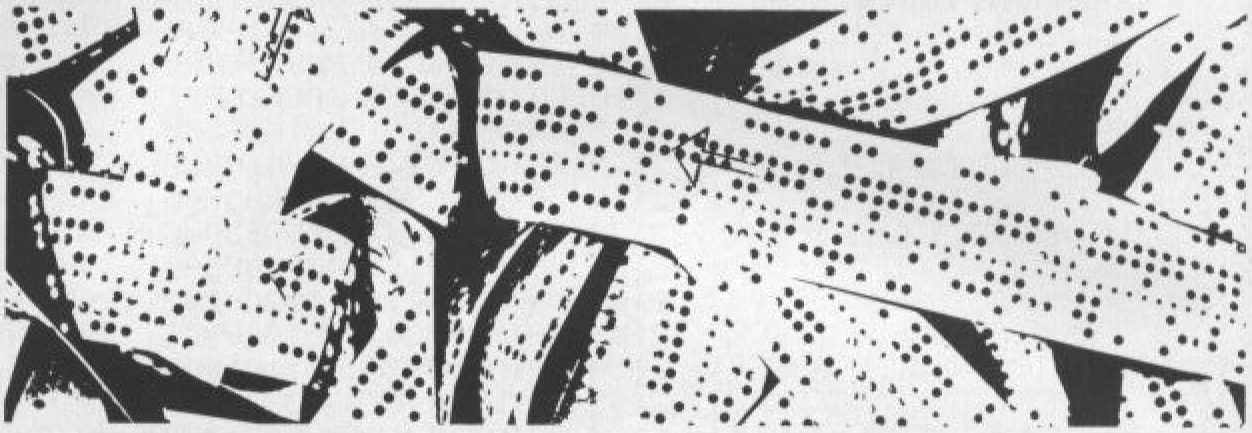
Program Listing

```
1795 DATA18690385FAA5FB69
1800 DATA0085FBA5F838E903
1805 DATA85F8A5F9E90085F9
1810 DATAE8E00AD0D060ADFF
1815 DATACF186901C9FFD001
1820 DATA608DFFCFADF9CF18
1825 DATA6901C93AD005EEF8
1830 DATACFA9308DF9CF60AD
1835 DATA0ED0186908C9F0D0
1840 DATA01608D0ED0ADF1CF
1845 DATA4AD005EEF2CFA980
1850 DATA8DF1CFEEF5CF60AD
1855 DATA0ED038E908C928D0
1860 DATA01608D0ED0ADF1CF
1865 DATA0AD005CEF2CFA901
1870 DATA8DF1CFCEF5CF60AD
1875 DATA0FD0186908C9F2D0
1880 DATA01608D0FD0ADFCCF
1885 DATA1869288DFCCFADFD
1890 DATACF69008DFDCFACF2
1895 DATACFC8C8C88CF2CF60
1900 DATAAD0FD038E908C942
1905 DATAD001608D0FD0ADFC
1910 DATACF38E9288DFCCFAD
1915 DATAFDCFE9008DFDCFAC
1920 DATAF2CF888888CF2CF
1925 DATA60A99320D2FFA906
1930 DATA8D20D08D21D0A234
1935 DATABD23C79D0706CAD0
1940 DATAF7A200A9019D08DA
1945 DATA9D30DAE8E028D0F5
1950 DATA6005A2018EEFCF4A
1955 DATAB005A9018DF0CF60
1960 DATA2041C2A5FC8DF5CF
1965 DATAA5FD8DFCCFA5FE8D
1970 DATAFDCFADFFCF85F8A9
1975 DATA4085F92086C1ADAB
1980 DATAC18DF3CFADACC18D
1985 DATAF4CFA9008DF2CF60
1990 DATA2016C520CDC120D8
1995 DATAC8ADFFCFA20420E7
2000 DATAC120F6C120A7C24C
2005 DATAA6C820C3C34C33C6
2010 DATA202BC34C47C6203C
2015 DATAC34C47C62040C34C
2020 DATA47C64CF6C820DFC8
2025 DATA4C47C6EAEAEA20E9
2030 DATAC84C47C620D8C8AD
2035 DATAFFCF85F8A94085F9
2040 DATA2086C1A000A90091
2045 DATAFAC8C040D0F94C47
2050 DATAC62022C34C47C620
2055 DATA06C74C47C6EAEAEA
2060 DATA4C47C62015C74C47
2065 DATAC6EAEAEA4C47C6EA
2070 DATAEAEAAADFCCF85F8AD
2075 DATAFDCF85F9ACF5CFB1
2080 DATAF8C951D013A95B91
2085 DATAF8A5F91869D485F9
2090 DATAA90B91F820EAC660
2095 DATAA95191F8A5F91869
2100 DATAD485F9A90191F84C
2105 DATAD4C6ADF3CF85F8AD
2110 DATAF4CF85F9ACF2CFB1
2115 DATAF84DF1CF91F86000
2120 DATA20B2C64C47C6AD8D
2125 DATA02C901D0042057C5
2130 DATA602037C560AD8D02
2135 DATAC901D00420A0C560
2140 DATA2077C560100C0103
2145 DATA0520020C010E0B20
2150 DATA0301131305141405
2155 DATA20090E2003320E20
2160 DATA2620101205131320
2165 DATA272A2720140F2003
2170 DATA0F0E14090E15052E
2175 DATA0405060317211133
2180 DATA02073C282B0DFFFF
2185 DATA9150565C62656B74
2190 DATA97A300304A54FFFF
2195 DATAC6C6C6C6C6C6C6C6
2200 DATAC6C6C7C6C6C9FFFF
2205 DATAA0A0A0A090928593
2210 DATA93A0818E99A08B85
2215 DATA99A0948FA0939481
2220 DATA9294A0A0A0A0A21E
2225 DATABD87C79D0706A901
2230 DATA9D07DACAD0F260A2
2235 DATA1EBDFF0349809DFF
2240 DATA033D070649809D07
2245 DATA06CAD0EDA018A200
2250 DATAA5C5C940D009CAD0
2255 DATAF788D0F24CB7C74C
2260 DATA07C878A5F88D1403
2265 DATAA5F98D15035860CE
2270 DATAEDCFF0034C31EAA9
2275 DATA048DEDCFAD15D049
2280 DATA808D15D04C31EA20
2285 DATA00C6204DC920F6C1
2290 DATAAD15D009808D15D0
2295 DATA78A9EF8D1403A9C7
2300 DATA8D1503584CA0C8EA
2305 DATAA08E8F928D818CA0
2310 DATA8DAD838F8C8F9592
2315 DATA131012091405200D
2320 DATA0F0405203A20A08E
2325 DATA8F928D818CA02020
2330 DATA2010010705209E15
2335 DATA0D0205123A202020
2340 DATAA200AD1CD0F00EBD
2345 DATA30C89D46C8E8E008
2350 DATAD0F04C7BC8BD28C8
2355 DATA4C6AC8A228BD37C8
2360 DATA9DFF03A90A9DFFD7
2365 DATACAD0F2A200BDF7CF
2370 DATA9D2504A90A9D25D8
2375 DATAE8E003D0F060EAEA
2380 DATAA5C5C940D010A028
2385 DATAA200CAD0FD88D0F8
2390 DATA2060C84CA0C8A200
2395 DATADD58C7D00DBD68C7
2400 DATA85F8BD78C785F96C
2405 DATAF800E8E00ED0E94C
2410 DATAA0C82060C84CA0C8
2415 DATA2000C62005C06020
2420 DATA05C020CAC420A7C2
2425 DATA602005C02019C420
2430 DATA4DC420A7C260A900
2435 DATA8D15D08D10D0A906
2440 DATA8D21D049088D20D0
2445 DATA8D860278A9318D14
2450 DATA03A9EA8D150358A9
2455 DATA9320D2FF4C32C920
2460 DATA00C0A9378D5AC1A9
2465 DATAC88D5BC1A90A8D60
2470 DATAC160A9678D5AC1A9
2475 DATAC18D5BC1A9018D60
2480 DATAC1A90085C6A9ED8D
2485 DATA28034CE8C3201FC9
2490 DATA20A7C26078A9318D
2495 DATA1403A9EA8D1503A9
2500 DATAED8D280358A9008D
2505 DATA10D08D15D020C9C5
2510 DATAA200BD080649809D
2515 DATA0806BD300649809D
2520 DATA3006E8E028D0EBA0
2525 DATA20A200A5C5C931F0
2530 DATA09CAD0F788D0F24C
2535 DATA70C9A99320D2FF20
2540 DATABEC9A000A200CAD0
2545 DATAFD88D0F8A9678D5A
2550 DATAC1A9C18D5BC1A901
2555 DATA8D60C14C50C3A900
2560 DATA85F8A93985F9A901
2565 DATAAAA820BAFFA908A2
2570 DATAF1A0C320BDFFA2BF
2575 DATAA03FA9F820D8FF60
2580 DATAXX
2600 REM ****************
```



This 64 utility from Mike Roberts will really speed up your tape system.

FAST TAPE SEARCH



THE COMMODORE TAPE system is notoriously slow. So is the disc drive for various reasons. This leaves the Commodore 64 owner with a bit of a problem. However, if you could find a program on a tape faster it would solve a lot of problems.

First on

My program is designed to be recorded as the first program on a tape. The procedure when wanting to load a program is to rewind the tape to the beginning, load and run the directory program but, and this is important, DO NOT PRESS THE STOP/ EJECT BUTTON. When the program runs it will ask you what file you want, and whether you are going to save to it or load from it. Next comes the important bit — you must hold down the PLAY key and press down the F/FWD key at the same time so that they are both down at once, now you must hold down the F/FWD key and press the STOP/ EJECT key. The end result of all this manoeuvering is a silent tape recorder and a depressed F/FWD key.

Now if you merrily press the Return button, the tape recorder will magically start to make a whizzing sound. After a number of seconds, depending on which file you want, the tape recorder will stop and you can proceed as normal.

This system will find programs on tape a lot faster than any normal method.

But wait. . .

There are one or two restrictions though:

The tape must be blank when you start to save programs on it, you can't just tack the catalogue on the beginning of a tape of programs and hope it will find them. The program expects the programs at evenly spaced intervals, that's how it works. The program is currently set up for 10 files of about 8K each on a C60 cassette. These parameters can be easily changed to suit your own tastes.

The number of programs on the tape is stored in a DATA statement in line 230. This must be equalled by the following number of DATA statements. The DATA statements are padded out with spaces so that if you add a new program name to the list and re-record it at the beginning of the tape it will be the same length as it originally was and not overwrite the first file.

The length of the sectors (currently 8K or 6 minutes) is dictated by the '10' in line 160. Double it for 16K sectors, halve it for 4K etc.

This is an invaluable utility and I have been using it on the PET, VIC, and now the '64 for about the past five years.

Program Listing

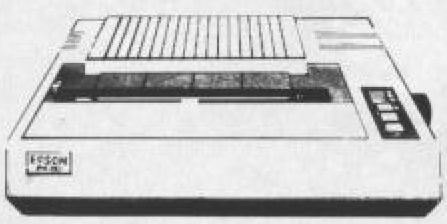
```
10 PRINT"]"
20 READX
30 DIMC$(X)
40 FORI=1TOX
50 READC$(I)
60 PRINTC$(I)
70 NEXTI
80 PRINT:INPUT"▒▒READ OR CREATE PROGRAMME/FILE";R$:R$=LEFT$(R$,1)
90 INPUT"▒▒WHICH PROGRAMME / FILE -- I.E 1,2";WP
100 IFWP=1THEN190
110 POKE1,39
120 PRINT"▒▒PRESS F/FWD AND HIT RETURN WHEN READY"
130 GETA1$:IFA1$=""THEN130
140 POKE1,7
150 T=TI
160 IFTI<T+(10*60*(WP-1))THEN160
170 POKE1,39
180 PRINT
190 IFR$="R"THENPRINT"▒▒HIT STOP/EJECT AND LOAD AS USUAL"
200 PRINT:IFR$="R"THEN230
210 IFR$="C"THENPRINT"▒▒HIT STOP/EJECT"
220 PRINT"▒▒IT IS READY TO SAVE NEW PROGRAMME/FILE"
230 DATA10
240 DATA"PROGRAM 1         1"
250 DATA"PROGRAM 2         2"
260 DATA"PROGRAM 3         3"
270 DATA"PROGRAM 4         4"
280 DATA"PROGRAM 5         5"
290 DATA"PROGRAM 6         6"
300 DATA"PROGRAM 7         7"
310 DATA"PROGRAM 8         8"
320 DATA"PROGRAM 9         9"
330 DATA"PROGRAM 10       10"
340 DATA"                   "
350 END
```

ORDER TODAY
PRINT TOMORROW
24 HOUR DELIVERY

SUMMER MADNESS SALE FROM SCI(UK)

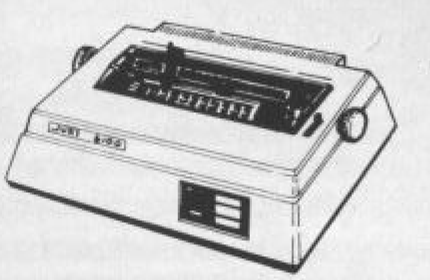
OPEN 7 DAYS A WEEK

EPSON PRICE SPECIALS



EPSON RX80 (DOT MATRIX)....~~£249.00~~ £199 + VAT = £228.85
EPSON RX80FT (DOT MATRIX).~~£285.00~~ £229 + VAT = £263.35
EPSON FX80 (DOT MATRIX)....~~£438.00~~ £324 + VAT = £372.60
EPSON MX100 (DOT MATRIX)..~~£475.00~~ £355 + VAT = £408.25
EPSON RX100 (DOT MATRIX)...~~£450.00~~ £385 + VAT = £442.75
EPSON FX100(DOT MATRIX)...~~£569.00~~ £499 + VAT = £573.85

DAISYWHEELS...*at an incredible new* LOW PRICE!



JUKI 6100......*just* £329 + VAT = £378.35

20CPS: BiDirectional & Logic Seeking
10, 12, 15 & Proportional Spacing
Wordstar Compatable
2K Buffer: 13 inch Platen
Underline: Backspace + Lots more
Centronics Interface Standard

THE DAISYWHEEL THAT HAS NO COMPETITION

OPTIONAL RS 232 TRACTOR AND SHEET FEEDER

We will match any Genuine Price Advertised—
SCI(UK) IS NEVER BEATEN ON PRICE



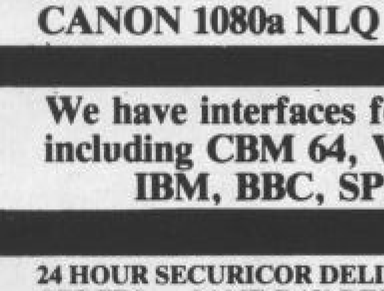
MANY MORE PRINTERS AVAILABLE - 1000s of SCI(UK) BARGAINS
send now for the **FAMOUS SCI(UK) Catalogue**



for the cheapest prices telephone **0730 68521** *or* **0730 68522**

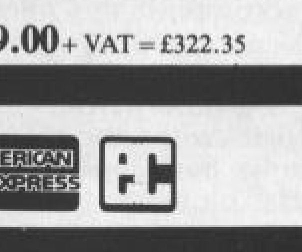
MORE SCI(UK) BARGAINS

SHINWA CP80....£179.00 + VAT = £205.58



Friction and tractor feed as standard
80cps
Bi-directional logic seeking
13 x 9 dot matrix giving true descenders
Sub and superscripts.
Italic printing and auto underlining
Condensed, emphasised, expanded and double strike printing (can be mixed in a line)
Parallel interface fitted as standard

FIDELITY 14" COLOUR MONITOR & COMPOSITE VIDEO
£189.00 + VAT = £217.35



New from the world famous CANON Company
CANON 1080a NLQ DM best value ever at..............................£289.00 + VAT = £322.35

We have interfaces for all types of computers, including CBM 64, VIC 20, APPLE, TRS 80, IBM, BBC, SPECTRUM, QL, etc.

Access MasterCard AMERICAN EXPRESS

24 HOUR SECURICOR DELIVERY £9.50 plus VAT • BANKERS ORDERS, BUILDING SOCIETY CHEQUES, POSTAL ORDERS — SAME DAY DESPATCH • ALL ORDERS COVERED BY THE MAIL ORDER PROTECTION SCHEME NATIONWIDE MAINTENANCE CONTRACTS ARRANGED • EDUCATIONAL DISCOUNTS VERY WELCOME
It's SUNDAY - Do you realise you can order NOW - We are open 7 Days a Week.

RANGEKEY LTD. Trading as SCI(UK)

DEALER ENQUIRIES WELCOME WRITE FOR DETAILS

SCI(UK)

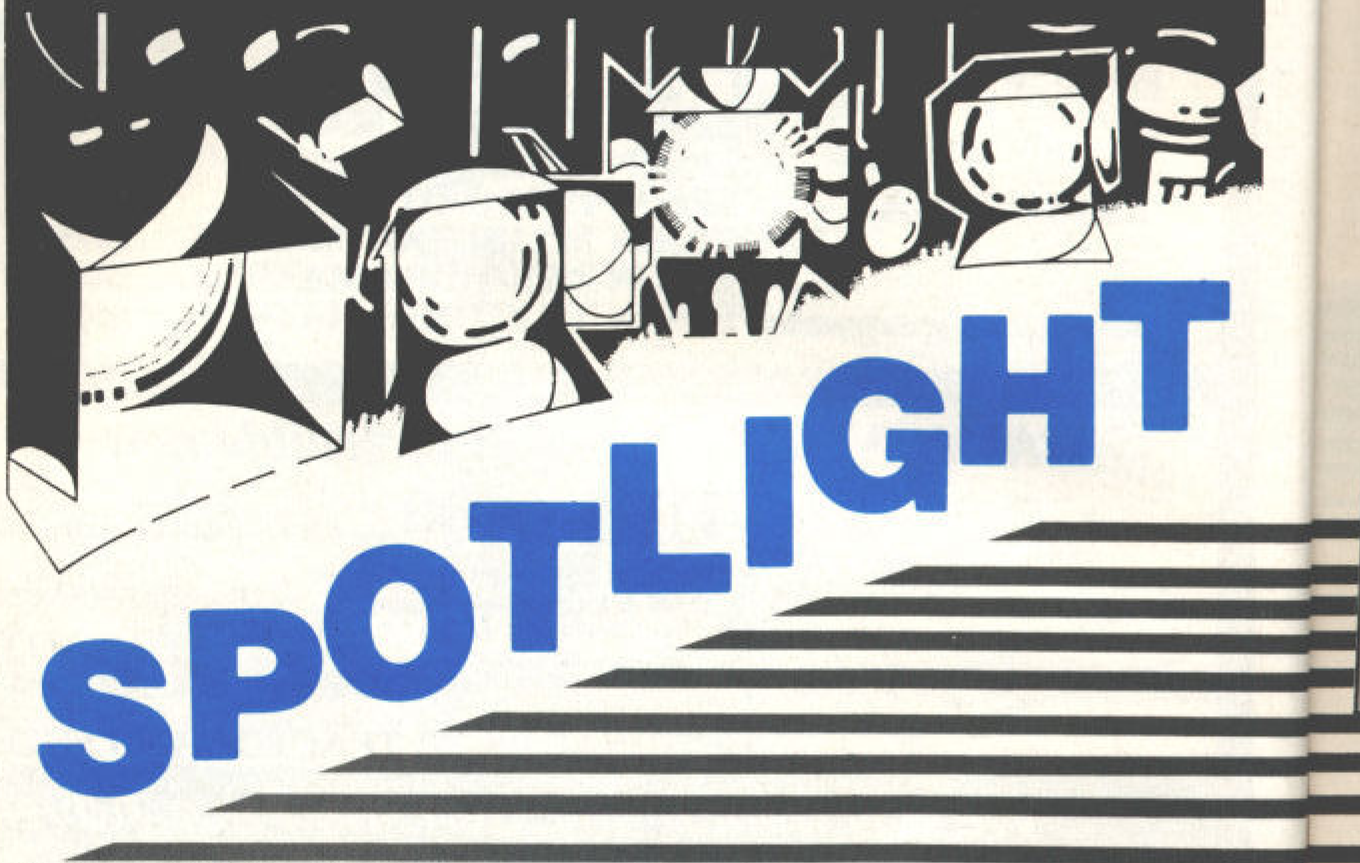
SCI (UK) FREEPOST (No stamp needed) PETERSFIELD HANTS GU 32 2BR

0730 68521
0730 68522

EXPORT ENQUIRIES NO TAX DELIVERY AT COST

Large showrooms now open at 12 High Street, PETERSFIELD, Hants. GU32 3JG Telex 86626 MYNEWS G

SOFTWARE SPOTLIGHT



We've searched the software shelves and thrown the spotlight on some available packages. See what our intrepid reviewers think of them before you part with your money.

OUR REVIEWERS HAVE spent long hours poring over the software to be assessed. They have tried to bear in mind a number of factors that will be of interest to you, the reader, such as the use of sound and graphics, how quickly the game became boring or addictive and to give you an overall impression of what to expect should you decide to spend some of your hard-earned cash.

Each review has been given a star rating out of a maximum of five stars (meaning brilliant, why haven't you rushed to your local store already?) down to one star (which means that the chances are you won't be ecstatic with your new purchase!). Always read the review and do not judge the package solely on the star rating — you may not be too bothered about the poor graphics and sound that have earned the package this opinion. And always remember that the review is only one man's opinion — one man's meat, etc. . . .

Encounter

★ ★ ★ ★
NOVAGEN
£9.95
CBM 64 + Joystick

SHADES OF BATTLE ZONE this one! Confronted with a 3-D plain (yes, it is 3-D), you hear the aliens approaching. Diamond shaped, fast moving aliens which don't fire at first, appear and are easy prey. Then, if you have the sound up on your TV, you will hear at times a low humming noise. This is when your shooting accuracy counts. The noise signifies that there is a suicidal alien missle coming towards you. One shot is enough but it's getting that one shot that is the problem.

Once you despatch the aliens a door or vortex opens on the screen. When you travel through it you will have to steer round a star field making sure you don't hit anything. When you accomplish this minor (!) task another door opens onto the second level. This level is a little harder than the first with the aliens shooting back and aliens which explode on their own.

Definitely a good game, having eight landscapes and three skill levels, it is very challenging. Having the sound effects is quite helpful as they tell you what to expect from simple aliens to a blipping exploding alien on level four.

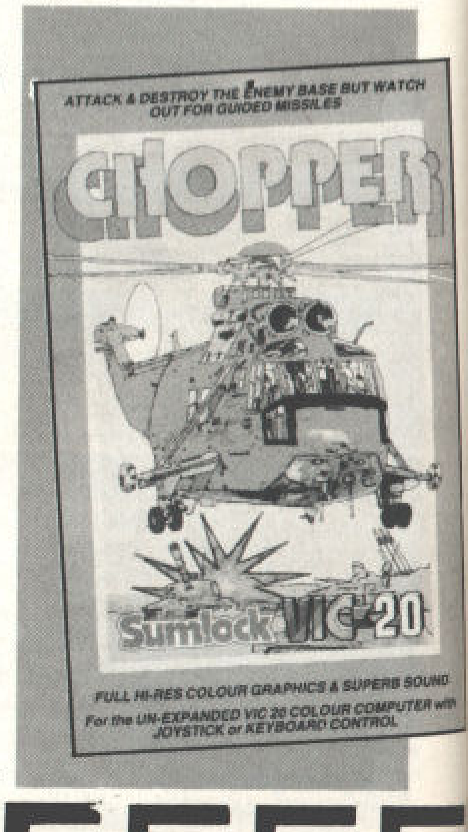
S.L.F.P.

Chopper

★ ★ ★
Sumlock
£7.95
VIC 20 + Optional Joystick

YOU HAVE ENTERED THE mountain ridge in your helicopter determined to stop the trucks carrying the arms from getting through to the enemies' base. They have heat seeking missiles which they fire at you. This is the setting from Sumlocks game, Chopper. This is done by moving your chopper left, right, up and down, dropping bombs on the trucks.

The graphics are jerky, the sound is not very good and the animation is fair. Loading is quite simple. It is an addictive game and very quick. The action is fast and it gets faster as the game progresses. It's a pity there are no skill levels. There are other good points especially the feeling you have when you have outrun the guided missiles. The scoring is very simple, 50 for a large and small truck, 100 for a tank and 250 for a missile launcher. The key choice is good and joystick response is also good. This is a brilliant idea for a game, however it's a great pity that it was not thought out more constructively. Overall a fast and reasonably enjoyable game from Sumlock but for £5.95 it doesn't appeal to me as value for money. P.W.W.



Maziacs
★ ★ ★
DK'TRONICS
£6.95
CBM + Joystick (Optional)

THE IDEA OF THE GAME IS to go into a maze and collect the treasure in the smallest possible amount of moves. Preventing you from doing this are perhaps the meanest bunch of nasties a video gamer could hope to run up against, the Maziacs.

The game starts with your character clutching a sword ready to do battle with the Maziacs for the treasure, which is encarcerated somewhere in the maze. To enable you to fight, you have to be strong and to keep your strength up you will find there is food around the maze. You will also find prisoners in the maze who are willing to divulge the whereabouts of the gold. The other way to find the treasure is by pressing 'V' which will display a portion of the total maze.

When you fight with the Maziacs you will find that you lose your sword if you kill it. This isn't a problem as long as there is a spare sword lying about, which there usually is. If there isn't then you may have to fight with your bare hands and then you stand a chance of losing your life.

It is worth paying attention to the graphics in the fight because of the intricate detail that has been put in. It is a good game with no two mazes the same.

S.L.F.P.







Skramble
★ ★ ★ ★ ★
RABBIT SOFTWARE
£5.99
CBM 64 + Joystick (Optional)

HAVING SEEN VARIOUS software companies make attempts at the game "Scramble" and being honest, not succeeding, I was quite surprised and excited at seeing this version.

Simply, it is even better than the original arcade machine game (yes, it's possible). To start with, the quality of graphics used in this game surpasses all other imitations. The game sequences are kept as close to the original as possible with two exceptions.

The first is that when playing you will notice some musical notes scattered on the floor of the landscape. Shoot these and the music is turned on or off. This is a good idea, since the only drawback is having "Star Wars" playing in the background. This spoils the sound effects.

The second deviation from the original is screen 3. This screen contains mobile tanks that shoot at an angle, and it's quite difficult to fire at them and maintain a good supply of fuel.

It is as hard, if not harder, than the original but the only high score feature is when you're playing. It's definitely well worth the money as it is an example of State of the Arts game. I look forward to more projects from Nigel Rowlan of Rabbit Software.

S.L.F.P.

Triad
★ ★
LIVEWIRE
£8.95
CBM 64 + Joystick (Optional)

TRIAD COMES IN A BOX containing no end of instructions, feature listings and pictures of the game. I couldn't wait to grab my joystick and start playing. On the box it said there was "3D total perspective graphics", "Smooth multicolour graphics, music and sound effects" etc, etc.

The game itself took a long time to load but I could wait! The description of the game was that the galactic merchants had constructed time routes in which they could travel. Unfortunately, the Triads kept on attacking without warning.

This is where your fleet of pilots come in. It is your job, firstly, to clear a patrol of Triads who advance on you at an alarming rate (on the easiest level) and, secondly, to clear a path through the meteors. Once both screens are accomplished, you see a merchant vessel disappear behind a planet and land safely, then you return back to the first screen again!

The graphics on this game are not exactly the best, although the laser sight is quite good as it is shown in perspective as you move.

S.L.F.P.

Alpha Blaster
★ ★ ★
Sumlock
£5.95
VIC 20 + Optional Joystick

THIS NEW GAME FROM Sumlock puts you in control of the latest space fighter; it is a version of the arcade game Astroblaster. This version has no speech or warp factor. The basic idea is to shoot down everything in sight. This includes spaceships of different kinds and sizes. They do come down in a uniform array of movement and you have to dodge the unshootable asteroids. After going through the asteroids you dock into the mothership; this is not a hard task because the computer automatically docks you. This is the basic idea of the game.

The sound is very good and the spaceships are in different colours. It is a fast demanding game. A novice at shoot'em up games is in for a hard time and even an expert won't find it easy.

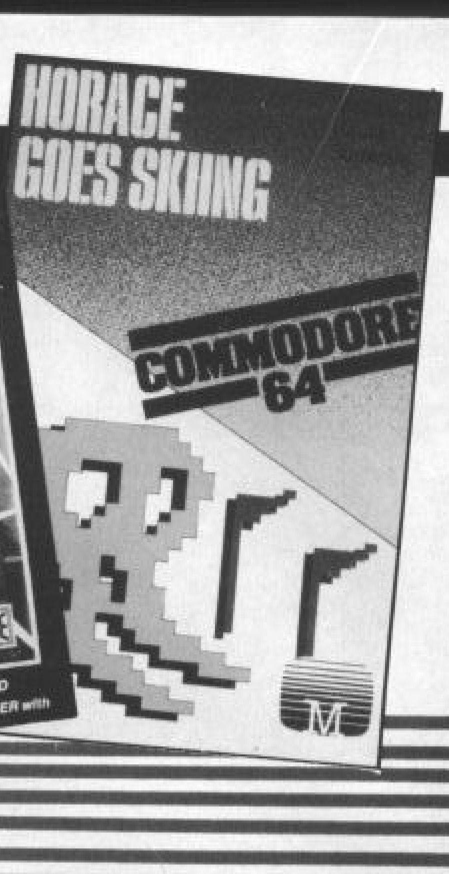
The key choice, Z right C left and Shift to fire, is good and makes it easy to play.

My only criticism is one bug. Overall this is a very good, fast moving game.

P.W.W.

SOFTWARE SPOTLIGHT





Skramble
★ ★
Sumlock
£7.95
VIC 20 + Joystick (Optional)

THIS IS A GAME IN THEIR 'Livewire' Series. Once loaded, which it did first time, the screen displayed the keys P,L,; and . For movement up, left, right and down shift to fire and CTRL to bomb. Key C starts the game. This is a version of the arcade game of the same name, graphics are fair and colour is reasonable. The game is written in machine code but is fairly slow, movements tend to be jerky, sound is used repetitively and continuously and becomes irritating. The idean of the game is to fly your spaceship horizontally through a series of caves, avoiding mountains, rockets, meteors and other hazards, using your lazers and bombs to destroy the enemy's missiles and fuel dumps. This is explained in detail on the colourful insert.

Considering this is for the unexpanded VIC 20 they have made good use of the memory. However, the end result is a game which is disappointing and soon becomes tedious.

It is quite a difficult game to play and the lack of a high score or hall of fame doesn't encourage you to keep trying to improve your skills.

P.W.W.

Metroblitz
★ ★ ★ ★
PSS
CBM 64
£7.95

Few basic zapping games seem to have the lasting appeal of good old fashioned Space Invaders. However, as a variation on the theme, Metroblitz will have you instantly hooked and as a piece of arcade action should have you practising in your sleep.

You are the sole defender of the metropolis which is under attack from wave after of kamikaze aliens intent on destroying the city below you. Your space ship is in perpetual motion and you have to prove your prowess by dodging between all the alien craft and obliterating them with your superior fire power before they destroy all the buildings.

Apparently there are twenty four waves of aliens to defeat and naturally points for every alien craft you zap plus an additional variable bonus at the end of each wave depending on how many you hit. At wave 13 you enter the advanced stage and that's when they home in on you as well as the city, pass onto the championship challenge at wave 19 and the termination attack at wave 22.

In short, it's fast, it's vigorous and it's energetic and I lost count of the variety of alien craft and the number of times it's broken my joystick. Whimper! K.M.

Horace Goes Skiing
★ ★ ★ ★
Melbourne House
CBM 64
£5.95

HORACE IS NO STRANGER when it comes to computer games. *Hungry Horace* was his first starring role. This time he's determined to display his prowess on the piste. Regardless of the quality of the game, one of the prime advantages is the use of a fast loading system which loads in less than two minutes. There's nothing worse than hanging around for ages waiting for a game to load! But I digress. Despite a partial similarity in the initial stages to those frustrating 'frogger' games, *Horace Goes Skiing* is an enjoyable and skilful game. The objective of the game is to get that cheeky little character across an extremely busy main road avoiding the juggernauts, cars and motorbikes and into the shed on the other side to pick up his skis. To reach the ski slopes of the Hannekon run you have to get Horace back across the road which gets busier as time passes. What happens if Horace gets splattered? Well it's not the end of the game. At the start of the game Horace has $40 in his pocket and every time he gets run over it costs him $10 for the ambulance ride. It also costs $10 for the skis so Horace has a few chances. Extra cash can be accumulated by continually crossing the road for which points are gained and a $10 bonus is given for every 1000 points. Once on the ski slopes the fun begins as Horace has to ski between the red and blue flags paralleling left and right and losing or gaining points according to his success. But there are obstacles. Strategically placed conifers get in the way and if Horace hits one, there is a chance he will have to start all over again if his skis break. There are also cleverly placed mounds to throw Horace off balance.

At the end of the run Horace simply has to cross the road again to reach the next slope. Of course there is more to this game than immediately meets the eye and despite its apparent simplicity has a high level of lasting appeal.

K.M.

Flight Path 737

★ ★ ★ ★
Anirog
£7.95
VIC 20 + 16K RAM + Joystick

THIS IS ONE OF THE ONLY flight simulations I have seen for the VIC and proved to be excellent. You have a choice of 6 levels, including a first solo flight (easiest) to test pilot (hardest). The game has an inflight clock, fuel indicator, altitude meter and many other dials found on the flight deck of an aircraft. On taking off you must have the flaps down and be at 180mph. The screen shows the runway which scrolls towards you in 3D perspective. The visual effects are good and are quite realistic. When airborne you see a backdrop of mountains. The graphics tend not to be as good as the runway but are effective. Whilst cruising you may have to extinguish a fire in the aircraft. This is done by pressing key E. You now come to the difficult bit, landing. It starts with ease but the element of difficulty soon appears when an adjustment of your instruments is necessary. Once landed, you are told how well you have, or have not, done and you are given a score accordingly.

The sound is very realistic and adds flavour to the simulation. This game loads very quickly and is one in a series of flight simulations to come for the VIC. I look forward to purchasing the next one out.

P.W.W



Star Commando

★ ★
Terminal Software
£7.95
CBM 64

STAR COMMANDO IS one of those games that looks like it will be an instant winner. But when I actually started playing there was something about it that was not quite right. I couldn't put my finger on it at first. Then it hit me slap bang between the eyes. The thing you'd expect to be able to control, the gunsight, was rigid and unmoveable and, instead, the joystick controlled the background. That was it. The game was ruined.

Nevertheless, onward in the face of adversity. As the rookie in the star fleet your mission is to repel the cosmic invasion force consisting of the Galactic Pirates and Sausonic Raiders and clear the space lanes of mines. Every sector of the galaxy has to be searched and cleared. The only problem is that your energy level is decreasing all the time and you need to find your mother ship to re-energise.

You do have an advantage, however, a long range scanner which can be consulted in any clear sector. The only problem is that the information is never complete.

Points are scored for each sector cleared and each invader blasted. The more points you score the higher your ranking at the end of the game — Captain, Commodore or Star Commando only skill dictates but the last does give you access to the secret code enabling you to claim your wings from Terminal.

Overall impressions: well nice concept, shame about the gunsight. K.M.

Cavelon

★ ★ ★
Ocean Software
£6.90
CBM 64

WHO SAID THE AGE OF chivalry is dead? Who wouldn't come to the rescue of a damsel in distress answering her pitiful cries for help? So you fancy your chances do you? Then enter the castle of Cavelon. The only problem is that it's not the easiest of tasks climbing your way to the top of the turret.

Castle Cavelon is a stronghold guarded by an army of knights and archers who wander the stone-walled corridors eager to deter any intruders. To reach the top of the castle turret you have to ascend six levels of stairs each of which is hidden behind a doorway at the end of a maze of corridors. To open the door you have to have all the pieces of the door in your possession. Search the corridors well because that is where they are hidden. Also floating around the corridors are a number of swords. Collect them all if you can. Their magic will render you immune for a while from the knights and archers. There's no doubt you'll need them all if you are to claim your prize.

On screen data indicates your points total and the comparative achievements of your rival if you chose the two player option, as well as the pieces of the door you have collected, the number of lives you have left and how many magical Excalibur swords you have managed to find.

You'll not be surprised to discover that this is yet another variation on the illustrious Pacman theme. However that is not to denigrate what it is. In fact, it holds its own well and is quite an enjoyable game. And anyway, who in their right mind would leave Guinivere crying in vain for assistance. K.M.

Flight Path 737

★ ★ ★ ★
ANIROG
£7.95
CBM 64 + Joystick

THANK GOD THIS ISN'T real. If it was, either the airline would have run out of undamaged passengers by now or my exploits would have been turned into one of those sky high Hollywood disaster movies.

Needless to say this is not one of the easiest games around but it has to be one of the more addictive. There's no giving up on it until what at first seems like the impossible has been achieved, even if it is only getting the jet safely off the runway and into the air without smashing the flaps to smithereens or breaking the undercarriage off.

The trouble is, that is only the beginning. Even in the easiest level there's the mountain to get over and then there's the problem of getting the plane down again safely on the runway without trying to tunnel beneath it. Like any flight simulator, Flight Path 737 requires a high degree of eye, hand, memory and general brain co-ordination.

As a representation of an aircraft flight deck, the screen is crammed full of flight information, heading, airspeed, altitude, fuel, flaps and under-carriage status and distance from home — including a view from the cockpit window.

Flying the 737 requires a combination of joystick and keyboard control plus a propensity not to panic. Just in case, by some freak chance, you manage to
Software spotlight 8
cope with your first solo flight, there are five other skill levels to tackle, all of which progressively shorten the runway and increase the height of the mountains, whilst throwing in the odd on-board fire and last minute crosswinds on landing. Getting your wings won't be easy, so whatever you do, don't panic.

If I had to criticise the game at all, and I feel really reluctant to do so, it would be on the basis that there is a tendency to master each level a stage at a time. If it's the landing you're always having trouble with, getting to that stage can be a bit of an unnecessary obstacle. The addition of a practice level would be a welcome solution.

K.M.





Mr. Mephisto

★ ★ ★ ★ ★
EURO-BYTE
CBM 64 + Joystick

TO BE HONEST, I DIDN'T know what to expect. Once loaded, using Euro-Byte's fast loader system called 'Overdrive', I was confronted by a cinema front with a 'Now Showing' sign including a piece of music the name of which is on the tip of my tongue! Very nice.

Now the game . . . Dave Lucas who is the Author of the game, even went to the extent of composing a poem in which there are clues to guide Hugo (that's you) through a tortuous, frustrating graphic arcade type adventure. I spent the first half hour, with beads of sweat on my forehead, trying to get past the first level!

The object of Level 1 is to get up the down escalator whilst trying to avoid the demons coming down from the top of the screen. You have to get through the door and that is not easy. I had to read the second verse for this screen (no more clues!)

Second level is a little more difficult, with yet more down escalators to go up! This time you have got three exits to go through. I went through the wrong one and had to start again . . . on Level 1!

It's a must for any computer game enthusiast who thinks he or she can play games well. It is well worth the money and you will be writing to Euro-Byte for more clues when you can't figure it out. S.L.F.P.

Marooned

★ ★ ★ ★
Buntasoft
VIC 20 + 16K RAM

THIS IS A 16K TEXT adventure. You have crash landed on the planet Paxar and the only way to get home is to find a missing part of your space ship.

The game is written in BASIC and with ingenuity you can break in and cheat. Even though written in BASIC, responses are fast and this is not a problem. The instructions are brief and typed into the insert. There is no sound except for the introductory tune. The text is black on white and in the caves, white on black. The program gives a desciption of the location including obvious exits and objects near by. Commands are given by usual verb noun combination although you can use one letter entry for N.S.E.W.U. and D. Pressing R gives a description of the situation that you are in. Pressing I gives you an iventory of the things you have, the game loads easily with a very snappy tune on the title page. The game itself is difficult but that is partly due to limited vocabulary and BASIC language analysis. The problems themselves range from easy to difficult. There is no save to tape option like on many VICventures.

There is a sense of humour. I found this adventure addictive. It caused me sleepless nights and many cups of black coffee.

P.W.W

Bongo

★ ★ ★ ★ ★
Anirog
£7.95
VIC 20 + 16K RAM + Joystick

A NEW GAME FROM Anirog, and one of the best I have seen. You are Bongo the super mouse, always wanted to be a mouse). You have to rescue the princess and win her heart, (I am definitely changing into a mouse). The only way to win the princess is to find the king's lost diamonds. Bongo sets out and eventually arrives at the thieves' hideout on the riverbank. This is where you take over. You control Bongo with the joystick which is very sensitive to the slightest touch. The game uses the whole of the screen; the graphics are very large and superbly designed. The game itself is in six parts: The layout is Kong/Manic Miner type; in other words, walkways, ladders and trampolines and an added bonus of a slides feature which only you can go down (obviously the robbers suffer from vertigo). There is a two player option and three levels. There are six screens which makes it addictive trying to get to the next screen. The sound is very good with a catch tune at the beginning.

It does take a while to load the game but it is well worth waiting for.

All in all an excellent game and should remain a firm family favourite.

P.W.W.



Multi Tron

★ ★
Sumlock
£5.95
VIC 20 + Optional Joystick

ANOTHER SHOOT'EM'UP from sumlock. A superb game which has several screens and is very fast. Your task is to penetrate the enemy defences and destroy all aliens at each stage. The first screen takes you onto the meanies; these you can shoot. The next screen is the warp stinger; these are very clever as they wait for your bullets to go past until they move. After this are the star-hoppers; these move diagonally down the screen and bounce off any obstruction except your bullets and you. The fourth screen are the tri missiles which come darting down until you hit them This is fairly easy as you can send up showers of bullets. The penultimate screen is by far the most difficult. Space turtles hide in their shells until, for a brief moment, they come out and you get the opportunity to shoot at them. At last you reach the last screen. The Cosmic phoenix's stand still and shower bullets down. The graphics on this screen are superb as the birds flap their wings. You are left with one final task which is to travel through the space corridor. This is easy and if you have managed to get this far you are granted a bonus ship. Back to screen one you are taken with an army of more furious aliens.

This really is a superb game and credit must be given to the writer's ability to fit all this into the unexpanded VIC.

P.W.W.



The Pyramid

★ ★ ★
FANTASY SOFTWARE
£6.50
CBM 64 + Joystick (Optional)

THE PYRAMID, LIKE MOST new pieces of software, comes complete with a high-speed loader to enable short waiting times for the player to play the game. Another nice aspect of this game is the abundance of instructions and information included. A high score form can also be found with the literature.

The program itself is about 'Ziggy' (that's you), making his way through 120 chambers of weird and wonderful creatures, all of which are dangerous to touch. The way to get through the Chambers is by collecting crystals when they appear, and dropping them on the forcefields which are guarding the next chamber. Ziggy can only pick up the crystals when they are in the harmless state. If Ziggy picks up a crystal before it is harmless then some of the shield power is used up. The shield power is always decreasing because of the aliens bumping into you.

If you reach zero energy before leaving the chamber, the game ends with you exploding and, assuming you score more than 30,000 points, your score is given a code which can then be sent into the software company. If your score is high enough, they will put you on their 1,000 top scores list published every 6 months.

Generally, it is a very good game with reasonable sound effects. S.L.F.P.

Plumb Crazy

★ ★ ★ ★
Terminal Software
CBM 64
£7.95

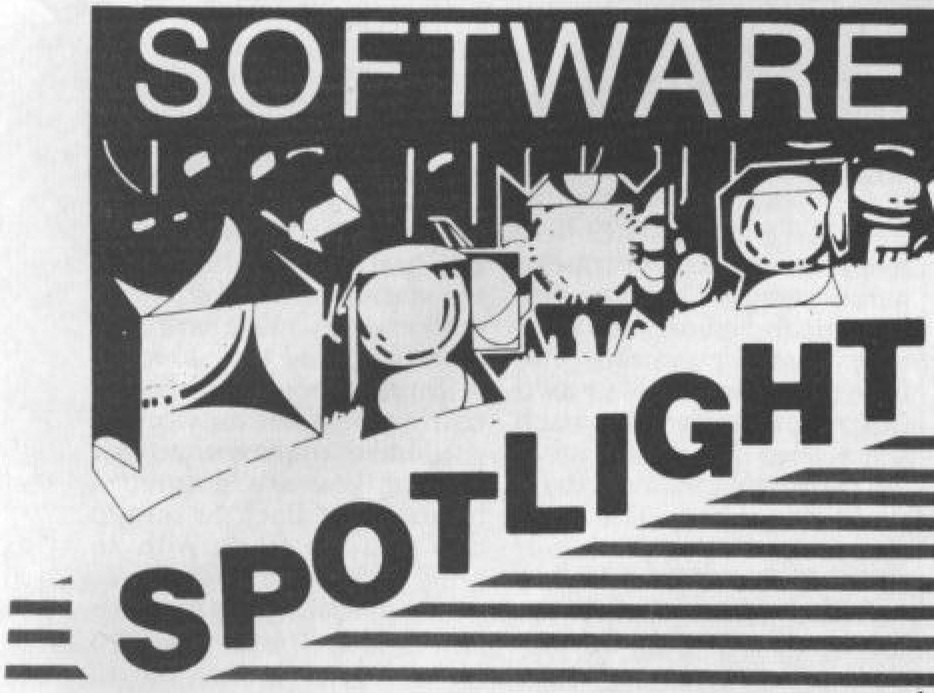
GOT A BIT OF A DODGY boiler have you? Fancy doing a bit of plumbing? Not really? Then why not get old George to do it for you. He's willing, one of the best drip fixers in the business and, what's more, runs around at your beck and call.

With the water temperature in the boiler rising rapidly and threatening to blow a stop cock at any minute, you have to get George to plumb a relief pipe from the valve at the bottom left hand corner of the screen to the boiler in the top right.

Easy? Well not quite. With time gently ticking away, you have to identify all the right bits of pipe at the bottom of the screen to weave the relief pipe around the boulders indiscriminately littered in your path. Mind you, if you do get stuck there's always that handy piece of dynamite to blast your way through.

But be on your guard. The ghost of George's former employer is on the rampage and dying to get him a bad name in the business by preventing him from completing the relief pipe in time.

Naturally the game gets progressively more difficult to the point where it is practically impossible to see the boiler for the boulders. So pick up your blow torch and get plumbing. K.M.



SOFTWARE SPOTLIGHT





Jumping Jack

★ ★ ★
Sumlock
£5.95
VIC 20 + Joystick Optional

THIS IS A VERSION OF THE arcade favourite 'Frogger. If you have not heard of it, the basic aim is to get a frog across a road, watching out for traffic. Once over the road you have to find your way to the lily pond, to get there you must hop onto turtles and logs making sure that you don't jump into the water. The obstacles gradually move faster making it quite fun to play. Alas in this version the graphics are very jerky and the animation leaves a lot to be desired. The choice of keys are good:. Forward,/ Back, Z Left and C right; this makes it very easy to play. The joystick movements are also very sensitive.

The game is written in full 6502 machine code. While the game is being played there is a 'tune' which soon becomes irritating; fortunately you can always turn it down.

Unfortunately there is a 'bug'. It is possible to go off the bottom of the screen and reappear at the top without having to cross over the road or jump onto the logs or turtles — therefore the game is much easier to complete and high levels can be achieved with great ease.

This is a dispointing version of the arcade game but is one of the few versions of frogger around for the VIC 20. It certainly won't wear out your trigger finger, but it does provide good family entertainment.

P.W.W.

Zodiac

★ ★ ★
ANIROG SOFTWARE
£7.95
CBM 64 + Joystick

ANYONE WHO HAS SEEN the game "Shamus" on the Atari will notice more than a passing resemblance with this offering from Anirog. As with most other games, there is a disruptive force on the loose. This time they come in the shape of Masters of Black Magic.

The Masters have stolen the twelve signs of the Zodiac and thrown them down in a 400 corridor maze, and it is your job to retrieve them and place the signs in the central room. But it is not as straight forward as that, because you have to fight the demon slaves which inhabit all the corridors. Be careful if you re-enter a corridor because there will be another set of demons to cope with.

If the demon catches you or you run into a wall, your character is immediately vaporised and all that is left is your hat!

Whilst playing this game I did come across some interesting features. The first was that after I had snatched a sign and had run into the next room, I found exactly the same sign again with more monsters! The second was a little bit more disconcerting. As my character ran from one corridor to another through a door, I materialised in a wall and got disintegrated yet again!

It is a good game with very good sprite graphics but it doesn't have that certain little something that makes it truly addictive.

S.L.F.P.

Chuckie Egg

★ ★ ★ ★
A & P Software
£7.90
CBM 64

WHAT A PLEASANT surprise! For Chuckie Egg read Donkey Kong. Enough said, I suppose. Do I need to go further? Perhaps I'd better.

You are Hen House Harry and the object of the game is to guide him around the hen house collecting the dozen eggs that have been laid. Yes, it's always a dozen and some are mystifyingly suspended in mid air. There is also a lot of corn to collect and all the while you have to keep old Harry out of the way of the ducklings which are busy gobbling up the corn and chasing Harry intent on pecking him to death. Why ducklings and not chickens I don't know!

Anyway, just like Donkey Kong there are platforms for Harry to walk along, ladders to climb up and down, and lifts to hitch a ride on providing you can get him to jump accurately.

There is a choice of six game speeds ranging from 'for those who prefer adventures' and 'for the faint hearted' through to 'for the general idiot' and 'for the suicidal maniac'. In addition, there are a variety of skill levels to progress through on each speed level but beware, on skill level nine the mother duck is let loose to run riot.

There's nothing at all pretentsious about this game. It's a good solid arcade action which follows a tried and tested formula which has proved very successful over the past year or so. But you need to know it for what it is otherwise, with the absence of any real description on the packaging, I suspect many a games addict would be disappointed in having bought something they already own but is branded with a different title. K.M.



Gridtrap

★ ★
Live Wire
£8.95
CBM 64

FIRST IMPRESSIONS ARE rarely wrong and in the case of Gridtrap it was pretty poor. However, before I get too negative I have to admit that the graphic representations are extremely good indeed. But, having said that, it is the execution of the game that really counts and that's where Gridtrap falls down miserably.

Despite some lengthy instructions, the game at first seems like a hopelessly overcrowded mess with a hideous musical accompaniment. It takes a while to fully realise what the hell is going on.

The object of the game is to manoeuvre a character called Mr Live Wire from box to box in the grid to reach and diffuse indiscriminately placed bombs. Points are scored depending on how quickly you reach the bombs and how many bonus scoring flagged squares you take in on the way. There are dangers of course. The skull and crossbones are out of bounds and several big boots skulk around waiting to kick you into touch. Once a square has been used it disappears from the screen although there is a facility to scroll the remaining squares in a line to help movement around the grid.

After every series of five bombs is defused, a new screen is presented and the game continues at a much greater difficulty level. Obviously the idea is to amass the highest score. At least the two player option brings an element of human challenge into the game which suffers dreadfully from trying to be too clever by half. K.M.

Bathtime

★ ★
PSS
£7.95
CBM 64 + Joystick (1 or 2)

THERE ARE GOOD SILLY games and bad silly games. The trouble is that I can't decide which this one is. It is very subjective as to whether you like violent or non-violent games. Bathtime is a non-violent game, which has reasonably good graphics and sound.

The main feature is its large use of sprite graphics including an elephant, a boy in swimming trunks, two heavenly cherubs, a goldfish and a swan. The game itself has an interesting concept, as follows: The fish and swan are having a bath and controlling the flow of water are the two cherubs; they, being you and another player, or the computer. It appears that the two cherubs have had a little argument and are dead set on making life difficult for each other.

Player one in the two player game runs the water into the bath whilst player two runs the water out. Player one has to either let the bath flood so that the swan sails away or let the bath run dry and kill the fish. Player two has to prevent this by monitoring the water flow. There is a time limit on each game played so as to let player two win if both swan and fish are still present.

To make things a little more difficult there is an elephant which keeps on appearing wanting a drink, and a boy replenishing from a bucket of water. This lowers and raises the water level respectively.

A good game for the very young but not for the space invader enthusiast. S.L.F.P.

Cybotron

★ ★ ★
Anirog Software
£7.95
CBM 64 (1 or 2 Joysticks)

SOME TIME IN THE FUTURE robots will become endowed with greater self-determination and will decide that humans are an unnecessary evil — exterminate the warm blooded ones. . .Fortunately Cybotron — a somewhat up-rated human — can save the human race. . .or can he (you!)?

There are several games with a similar theme on the market and although I can only get to level 7 with difficulty (there are 100!) there is the feeling of a reasonably balanced game — that with a little more luck (practice!), I could get further. Only joysticks may be used — no keyboard — but there is the option to use two. . .one for movement and one for firing at the 'baddies'. Who says computing is a solitary pastime? If you have time to assimilate the graphics you will find there are 6 different robots, 5 types of electrode (stationary objects that you can destroy but must not touch), the odd humna to rescue and of course your 'alter-ego', Cybotron. Hi scores are recorded and you have 3 lives to gamble with. Bonus waves, where there are a higher ratio of humans (to rescue) to robots will help boost your score — and to rest your shattered nerves, there is a pause option. Cybotron has 'Turbo Load' to cut down on loading time. . .and this worked perfectly every time. A game for the 'hunt and shoot' brigade with quick reflexes — well balanced and worth a try . . . P.F.

SOFTWARE SPOTLIGHT



Dr Watson computer learning series — Basic adventure part 1
Glentop Publishers Ltd
(ISBN 907702 16 2)
£9.95

THE 'SLEEVE' ON THIS product has several — let us be generous — ambiguous statements. Let us look at these and try and decide what we have in our hands.

First: "The revolutionary new concept in computer learning provides a FUN way for children to step into the 21st century world of BASIC programming. . ." So far, so good — the book does present a novel teaching concept. Programs and exercises are introduced in a storybook manner that might well catch a child's imagination.

Second: "Part One of the book is an exciting adventure. . .(our heroes) must learn to operate the ship's computer if they are to escape!" We now come to an ambiguous bit! The word adventure to most computer owners implies that the user has to solve an adventure — not so, the book works its way carefully through examples of PRINT, INPUT and String handling and in the last two pages of the story our intrepid Dr Watson jury rigs a spaceship's controls — with no reference to our previous examples of computer programming — and sets it in motion. Not qute what we expected!

Third: "This unique adventure is specially designed to teach the fundamentals of '64 BASIC by way of example — the way children learn best." Well. . .perhaps so — but for the use of that word 'adventure' again.

Fourth: "Accompanying this is a tape containing the same programs as are on the spaceship's computer, and some BONUS teaching programs so your children can learn as they play." OUCH! Yes there are three simple programs that are referred to in the book — very short but yes in the book! We then have two copies of a 'Hangman' program — a little old hat but I maintain that one can have a lot of fun with these sort of guessing games — BUT with a total vocabulary of four (4) words? AND one of those is CBM — not exactly inspiring, it also makes one wonder about who is kidding whom at the price of £9.95.

Having turned the tape over and LOADed the first program, we are presented with a menu. This lists the programs on this side of the tape and on selecting option 'x' you are given a very brief resumé of what you will learn. ..then there is a 'Message from Dr Watson' — this turns out to be pages of errata from the book! With my package there was no indication that this was there — not even a cheap slit of paper. . .

The teaching programs request the user to type in the correct format for PRINTing various examples — followed by similar exercises on String handling and use of Line Numbers — quite a nice idea and well though out. . .except. . .with a teaching program such as this one does not expect to be able to 'crash' the programs if given a wrong answer — OK, I tried to do it but . . .

Fifth: "Every BASIC command covered is also given a separate careful explanation in the second part of the book. . ." Yes it is . . .except for the errata mentioned above, it's passable.

In conclusion, all I can say is. . .wait for Mr Holmes's book — he always knew better than Dr Watson!

P.F.

Booga-Boo
★ ★ ★
Quicksilva
£7.95
CBM 64 (Joystick only)

THIS MUST BE ONE OF THE most frustrating games I have ever played for a long time! You are a flea (!) trying to get out of a cavern by jumping from ledge to ledge, to reach the exit above you. The 'power' of your jump is controlled by how long you hold the joystick left (or right) before centreing the stick — then you jump. It takes a little while to get the hang of judging how much power to put into a jump, in order not to overshoot or undershoot your target. Just to add to the excitement — there is a fierce dragon roving around the cavern, ready and willing to gobble you up! You have only one life and having lost that (I did, with monotonous regularity!), you have to wait through the 18 seconds 'intro' as Booga-boo falls into the cavern once again.

Time elapsed, Level and Hi Score are indicated at the bottom of the screen. Level is the height you are at, above the floor of the cavern. Hi Score does not indicate until you have escaped the cavern, so no matter how long you manage to evade the dragon, or how high you climb, you have no personal best score to strive for in the early stages. . .I could have done with a little something to spur me on after being gobbled up for the fiftieth time! Graphics are superb, I just wish it was a bit easier at the start, to help inspire me to greater heights! There is no pause option — but then if you don't last long enough to achieve a Hi Score, you can always drink your coffee in the 18 second wait following death!

P.F.

Typing Wizard

★ ★ ★
Severn Software
CBM 64

WHAT A SMASHING TITLE page — I always did like watching wizards throwing lightning bolts. . .out of the field of fire, naturally! What a pity, one of the menu options is not 'wizard recall'. This program sets out to teach you touch-typing — a very commendable project, especially as I don't! I would guess it was written using a monitor, as the choice of colours for some instructions was orange letters on a medium-grey background - not the easiest to read, even if you turn the colour off at the TV. Instructions shown like this are few and anyway you want to get on with your typing so perhaps Severn may be forgiven — a thought for the future could well be user definable colours?

Menu driven, you are offered nine options — 1. is the Introduction. From then on you are told how you should place your fingers and thumbs for different rows on the keyboard — starting with the 'home' row ASDF

This sort of program is really only as good as your 'will to learn' — if you have that 'will' . . . Typing Wizard can only help.

S.L.F.P.

Space Pilot

★ ★ ★
ANIROG SOFTWARE
£7.95
CBM 64 + Joystick (Optional)



PACKAGED IN THE NOW standard library case, Anirog's "Space Pilot" comes with a picture showing aircraft of different eras fighting each other.

In this game you are an ordinary 1984 Jet Pilot. The very fact that you are from that particular time in avionic development is a definite help on the first 3 screens. In the first screen you are faced with a 56 strong squadron of bi-plane fighters (the year being 1919). Easy you think, and easy it is, for these pilots do not stand a chance against your gleaming 1984 fighter. All that is left to do on that screen is to tackle a single Zeppelin which comes floating across your screen, but be careful because it can release a nasty little salvo of bullets when you are not expecting it. Once you have cleared the screen, you are transported to another time — 1940. Again, you must despatch a large number of aircraft. But they are getting a bit more nasty.

Clear that screen and you are launched into 1970 where you meet helicopters. This is the turning point of the game. As you clear that screen you are transported into 1990! The aircraft are more advanced than yours, so you have to play more tactically instead of just flying into them and shooting.

Finally, if you clear that screen you meet from the year 2001, Flying Saucers! Unfortunately I didn't clear that screen but I died valiantly!

This game in general has good graphics using sprites; the sound has a few effects but not too much to write home about. With the time limit and other obstacles like the clouds it can lead to very interesting suicide runs.

S.L.F.P.

Splat

★ ★ ★ ★
Incentive Software
CBM 64
£6.50

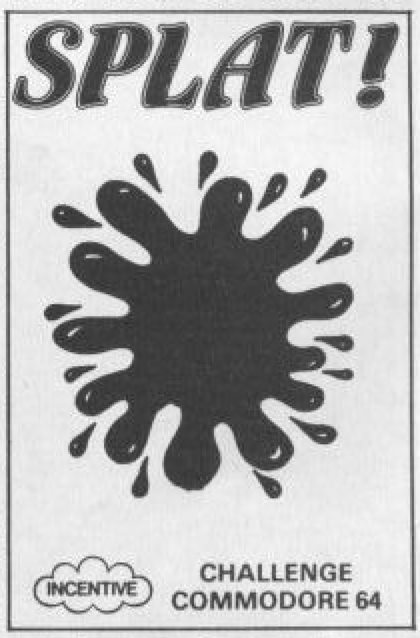
THIS IS A GAME FOR ALL potential members of the Strawberry Jam Preservation Society. The object of Splat, amongst other things, is to prevent the hero, Zippy, from being thoroughly pulverised or, more colloquially speaking, turned into strawberry jam.

To all intents and purposes, Splat is a variation on the age old theme of the maze game. Comparisons can be drawn with Pacman, the illustrious ancestor of all maze games, although it can be a bit misleading. However, if the truth be known, it is just as addictive.

The main difference between Splat and other maze games is that the main character, Zippy which you control, is not being chased by an nasties, at least none that I've managed to discover. Zippy has plenty of freedom of movement within the boundaries of the maze which is much larger than the visible playing area and slides around unpredicably intent on splattering Zippy.

The purpose of the game is to explore the maze, scoring points by gobbling up grass, plums and unstable thingermejigs which have a habit of turning the tables on Zippy. There are seven skill levels in all and you have to race against the timer keeping Zippy out of trouble long enough to reach the next level.

The main obstacle to avoid are the spikes which appear for the first time in level two, the water be-



cause Zippy can't swim and the sides of the playing area. There are also things called alternative Zippys which could be good, bad or just plain ugly but my failure to progress beyond level five has failed to reveal them. The lack of any useful instructions doesn't help matters much either.

The great thing about Splat is that reasonable skill levels are quickly achieved and it has a perfect frustration factor which makes you keep plugging away at the high score.

One thing that might help, but I'm not quite sure should exist, is slow motion movement when the pause facility is on. Not only will it help calm your nerves but can also help you get through the difficult bits.

K.M.

Fancy having a go at writing your own arcade type game? William Fong has tried to give you some guidelines to start you on the way.

YOUR OWN ARCADE GAME

HOW MANY TIMES HAVE you played a new computer game and after minutes (or sometimes hours!) of frustrated playing you are sure that you could have written a far better game yourself? It is really not as difficult as you might think provided you approach the task in the correct fashion — and you can get a great deal of enjoyment from the program writing itself, let alone when you can happily play with the finished product!

The first move

Naturally your first move is to think of the basic idea for the game. Of course the idea should be an original one — try to break away from the same old mould of alien-zapping, maze-wandering and monster-killing! No doubt this is the most difficult part of writing a good game.

Now don't immediately rush off to your computer — well not yet anyway! Transfer your idea to paper — there's nothing worse than spending hours at the keyboard only to find that the game falters at a crucial time — this is a fine recipe to make you give up on the idea before you've given yourself a fighting chance. Think everything out before you get too carried away: what sort of characters will be involved? Are you going to use graphics and sound in a particularly unusual way? What will signify the ending of the game? Do you want a two-player facility? These and so many other factors will be vitally important to establish before you go headlong into typing actual code.

Throwing down the gauntlet

Bear in mind that the game should be challenging to be even the most skilful player. This does not mean that the game should be impossible to beat — people will easily lose interest in a game where they haven't got even the faintest chance of winning! A way of getting round this problem could be the use of different levels of difficulty. Above all keep the concept simple — too many rules can be confusing, and they should remain the same throughout the game.

Most people prefer a joystick-operated game, but it is wise to keep all your options open. If you have not got a joystick and are forced to use the keyboard, please give some thought to the choice of control keys so that the player does not have to perform superhuman digital antics in order to fire a missile!

On screen action

Your own arcade game 2
Having got your ideas and plans on paper, you can



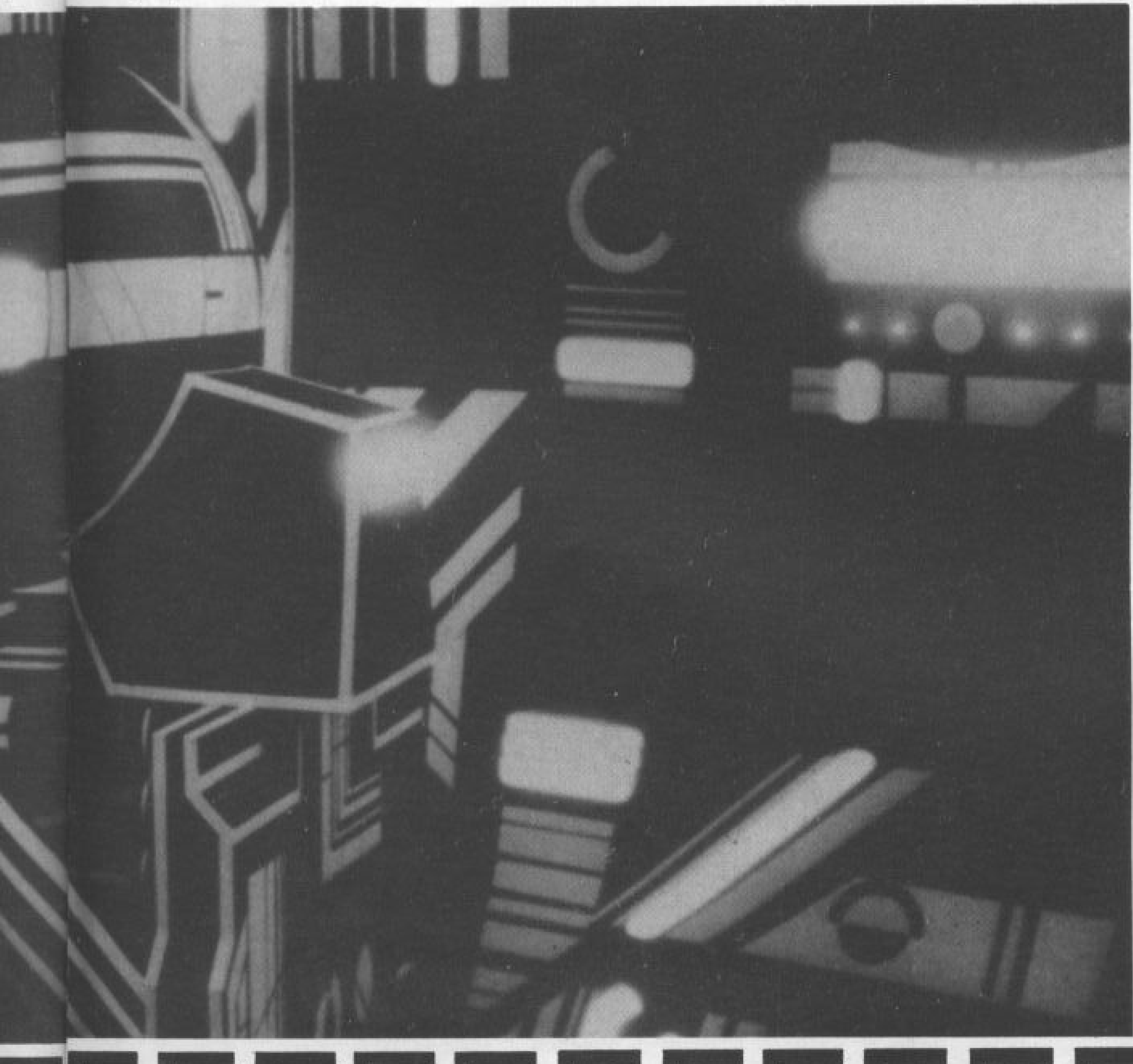
touching the computer. A lot of thought is needed to make the game interesting and addictive — this is often the stage at which programmers don't apply too much care and the small but important touches are left out. In good games the screen is always changing and the graphics are usually moving. At this stage you can let your imagination run riot when thinking of the representation of the main characters in the game and here you can use the CBM 64's graphics abilities to the full. But don't worry if you find it difficult — there are many graphics design packages around for the CBM 64.

Colour use is also very important and is probably best added after everything else has been completed. Generally colour should be used to minimise the player's frustration; critical features and anything that's urgent should stand out clearly, whereas features that are not viewed often should be lightly coloured. Be careful with the colours of adjacent characters as the eye is easily fooled by contrasts; try also to avoid filling the screen outside the playing area as this can be distracting and confusing. It is also often worth the trouble to try playing the game on a mono screen — different colours do not always distinguish themselves on a black and white TV.

Listening in

Sound effects should be considered carefully; certainly with the 64's complicated SID chip excellent sound effects may be generated. Good use of sound can enhance a game and turn it into something really special, but it is perhaps one of the most difficult things to perfect. It must suggest impending action to the beginner but not antagonise the advanced player; it is worth leaving in the option of turning the sound off. Trial and error is the only way to perfect different sounds for each character or piece of action — a long, low frequency sound suggests slow movement but you need a short distinct sound for more immediate action.

Try to make your sounds as pleasing as possible and try to avoid shifting from low to high frequencies too abruptly. If you are a little more expert in the musical field you could have a go at writing a little tune to go at the beginning of the game or running throughout it. Don't forget though that a catchy little tune heard multitudinous times while you're trying to beat the computer can sometimes make you want to beat the composer instead!



Other factors

Don't forget that throughout the writing of this wondrous game you will need to keep track of where the characters are at any one time — was the alien destroyed and therefore now out of the game, or have you merely blasted it into another position in the maze? There's surely nothing worse than having expertly destroyed a submarine only to find that minutes later it has mysteriously come back to life and suddenly blown your ship to smithereens!

A scoring strategy is purely a matter of personal preference, although it certainly appears that many people like a High-Score record facility. You must also decide whether to have a time limit on the game — will you run out of fuel, missiles or oxygen?

Game point

Well, that's all there is to it! I know it seems a totally daunting task, but rest assured (?) that the hardest game you will ever have to write will undoubtedly be your first. As you go through the routine for the first time you will gradually learn nifty little movement routines, exciting explosion effects and more, and often you will find that they fit in better to your Mark III version of the game which has all sorts of refinements added to the original. Throughout the process of writing your own game you must continually test each new stage you add on — who knows how that new routine will affect the part you have already perfected? It is also advisable to always keep back-up copies for those odd occasions when the whole thing crashes — you may not always have the stamina or memory to go back and start at the beginning again!

STUCK?

Try a Course prescribed by Dr Watson

DR WATSON series

BASIC Courses In Space For The Commodore 64 & VIC 20

An exciting course for all children from 7 to 77. Learn BASIC while enjoying this most exciting space adventure. Master the computer to escape the Alien Spaceship.

It combines a story which is compulsive reading for the young reader with a carefully structured approach to BASIC intricately interwoven into the text. Also, each BASIC command covered is given a separate, careful explanation in a special 'easy reference' section in the second half of the book.

The tape includes the programs on the ship's computer, and four computer aided learning (CAL) programs which teach the major BASIC commands.

"It should hold the interest of an under-12 very well."

– Home Computing Weekly.



BOOK & TAPE

£9.95

Beginner's BASIC For The Commodore 64

This project-based course guides you step-by-step through BASIC, developing games and utilities using progressively more complex BASIC commands. All the programs developed are provided on tape so that you can get a taste of them first, before working along with the detailed explanations given in the book.

Programs include three video ball games, a character generator utility, a sprite generator utility and a 'composatune' music utility. You also get 'Honey.Aid1', a machine-code utility that adds 28 commands to Commodore 64 BASIC allowing easier prgramming, sound and graphics.



BOOK & TAPE

£10.50

"The Dr Watson series is rapidly becoming recognised as quality material."

"... a well thought out tutorial... if this was the total content of the course, it would prove excellent value. However, there is more . . . much more."

" . . . the course consistently encourages you to learn more and more by making learning fun."

"other . . . manufacturers have been shown how an introductory manual should be written."

– Commodore Computing International.

Beginner's Assembly Language Courses For The Commodore 64, VIC 20 And PET

These courses introduce the real beginner to assembly language programming. No prior knowledge of assembly language is assumed and the aim is to ensure that *every* reader succeeds. Numerous examples illustrate the points while exercises along with solutions test the understanding.

The tape includes an assembler which assembles and disassembles code anywhere in memory. Also on the tape is a binary/BCD/hexidecimal CAL tutor program which teaches about the various mathematical notations used in machine code programming.



BOOK & TAPE

£12.50

"There may be easier ways of adding two and two together . . . But there aren't easier ways of learning how your computer does it."

– PCN.

"No self-respecting VIC 20 owner interested in programming should miss this instruction set."

"The manual would be worth buying for the reference section alone."

"Don't miss it".

– Which Micro.

"I have made more progress in a week than messing around for a year with other books and articles."

"a really super clear book . . . "

– The VIC 20 North London User Group.

" . . this book is worth its weight in gold."

"If there was ever a good beginner's guide in this field, then this is it."

– Commodore Computing International.

HONEYFOLD

HONEYFOLD SOFTWARE LTD
B1 Standfast House
Bath Place
High Street, Barnet
London EN5 1ED
Tel: 01-441 4130

Take to the air and fly with us in Concord. Fasten your seat belts tightly though and see if you can land the aircraft in William Fong's simulation for the 64

CONCORDE II



CONCORDE II

FOR MANY YEARS THE French and British airways have been discussing how to speed up travel and at last they have designed a new Concorde; Concorde II. This should be far better than the original Concorde as it travels much faster and is more economical to run. Being a 'professional' pilot yourself you have been given the chance to have the first test flight. Your route is to Glasgow from Heathrow. The journey is not as easy as it seems as there are quite a few instruments to watch and the level you pick will have to depend on the weather conditions you will encounter. The flaps, undercarriage, speed, altitude, deviation readout and artificial horizon are all important instruments to watch.

How to fly

When you take off make sure your flaps are down or the plane will not go up! When your flaps are down your speed must not exceed 280 mph or you will damage your flaps. Your undercarriage (wheels) must be taken up before the speed reaches 300mph or they will be damaged. However, the undercarriage must be down before the aircraft is below 300ft. If your flaps are up the plane speed must not drop below 210mph.

The distance between the two airports is about 380 miles and because this is a real-time based simulation it will take a long time if you keep the speed at 500mph. The aim of this test flight is to see how fast you can get to the other airport : speed approx 1700 mph.

The artificial horizon is an important instrument and it comes in very handy when visibility drops to zero. The deviation readout should be updated often as the pointer is in line with the approaching runway, so try and keep the pointer near zero. The map readout on the British Isles is useful as it indicates how far you are from Glasgow (if you know where Glasgow is).

Where nearing the runway or any other unknown information the control tower will report to you. When going in to land the speed must be about 200 mph and the altitude less than 60ft. The flaps must be down and the undercarriage must also be down. When you are above the runway drop your speed to 10mph and altitude to about 20ft then you will get a perfect landing.

Controls

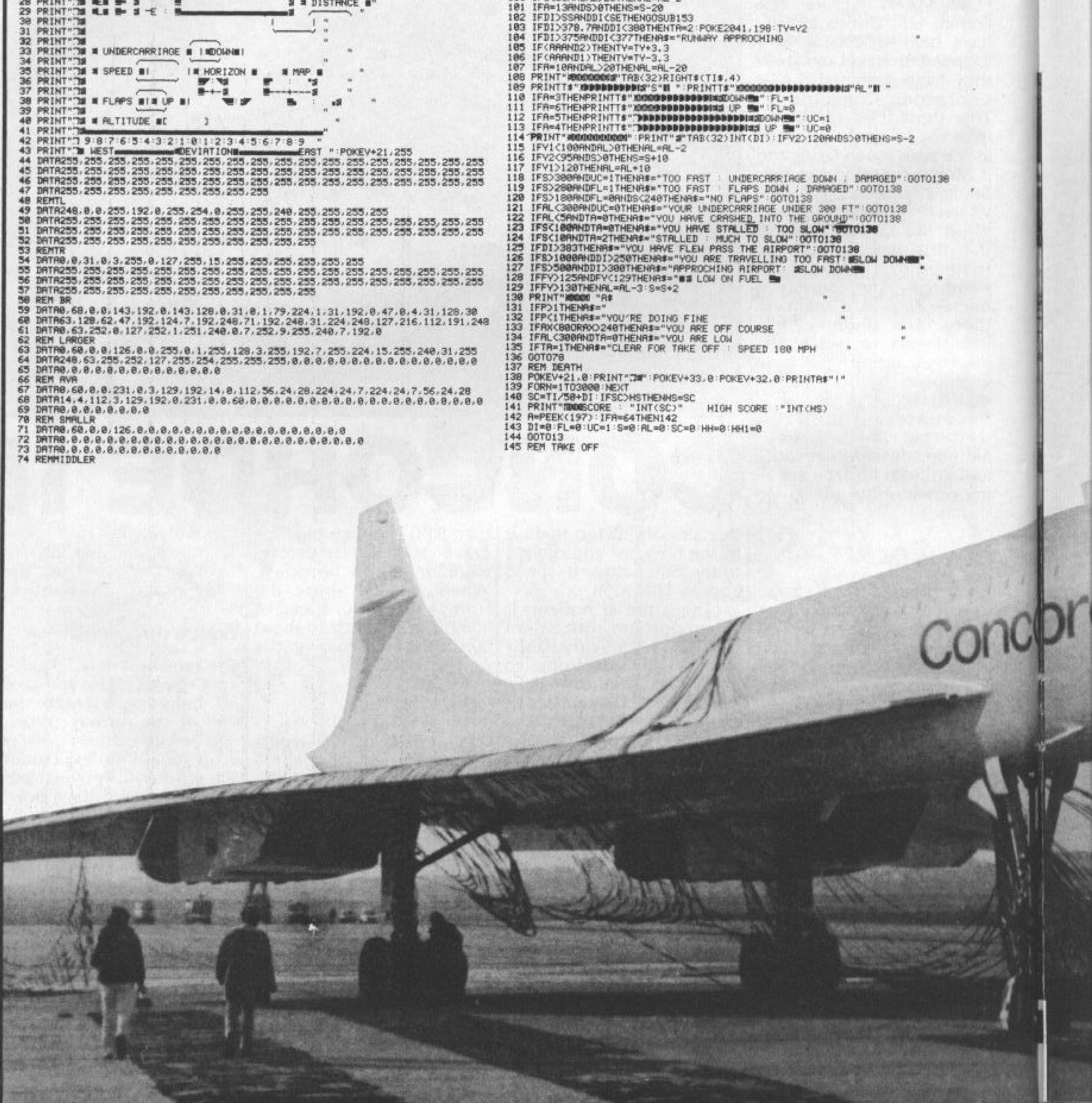
To turn push joystick in port two and turn as normal.
To increase altitude pull the joystick back
To nose dive push the joystick forward and the plane will dive down
To lower your undercarriage use Function key 3
To raise your undercarriage use F1
To lower flaps use F 5
To raise flaps use F 7
To drop altitude without the plane diving use "A"
To drop speed use "S"
To increase speed use the Fire button on the joystick.

Watch out

The runway is only 1.3 miles long so make sure you take off before you reach the end of the runway. When you see the other runway drop your altitude to about 50ft or you will fly over it and all effort would have been wasted.

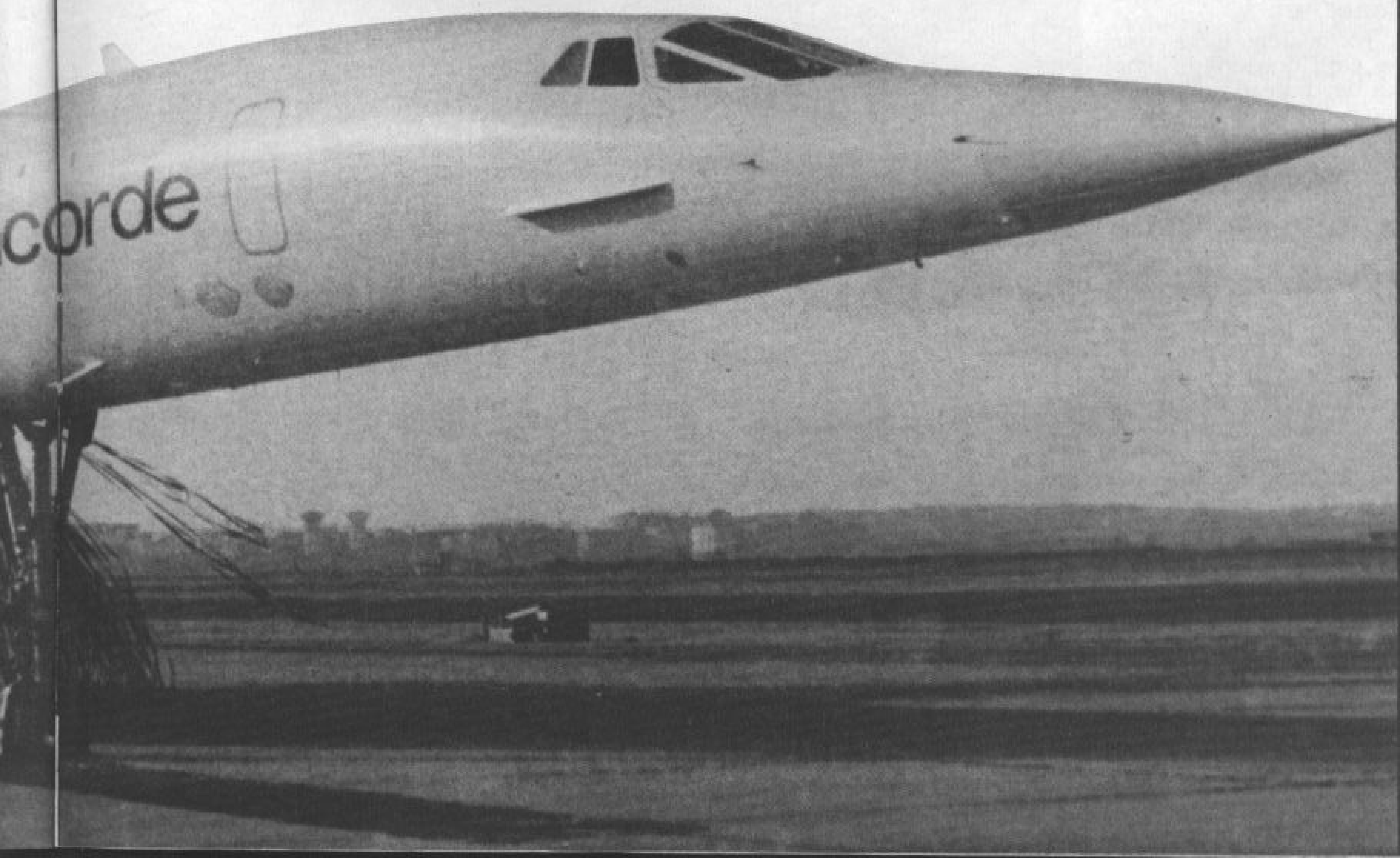
The game may sound easy to complete but wait until you are in the cockpit. There are levels you choose to increase the difficulty, storm, engine failure, strong wids, and many other conditions. Good luck, and happy landings.!

```
1 REM *** CONCORDE II : WILLIAM FONG **
2 POKE53281,0:POKE53280,0:PRINT"■"TAB(15)"PLEASE WAIT■■■■■■■"
3 PRINTTAB(15)"■CONCORDE II■■■■■■"
4 PRINTTAB(4)"■BY WILLIAM FONG : COPYRIGHT 1984"
5 FORN=0TO62:READA:POKE896+N,A:NEXT
6 FORN=0TO62:READA:POKE832+N,A:NEXT
7 FORN=0TO62:READA:POKE960+N,A:NEXT
8 FORN=0TO62:READA:POKE12480+N,A:NEXT
9 FORN=0TO62:READA:POKE12544+N,A:NEXT
10 FORN=0TO62:READA:POKE12608+N,A:NEXT
11 FORN=0TO62:READA:POKE12672+N,A:NEXT
12 FORN=0TO62:READA:POKE12736+N,A:NEXT
13 V=53248:X1=112:Y1=110:Y2=Y1:Y3=Y1:X2=160:X3=208:HX=176:HY=190:FX=40:FY=100
14 VO=54296:WA=54276:HI=54273:LO=54272:AT=54277:UC=1:FL=0:TA=1:TX=157:TY=107
15 W3=54290:H3=54287:L3=54286:A3=54291
16 TI$="000000":T$="■■■■■■■■■■■■■■■■":DI=0:TT=2:BX=244:BY=153:AY=223:SO=1
17 POKEV+23,230:POKEV+29,230:POKEV+27,254:POKEVO,15:DT=19:AX=INT(RND(1)*100)+50
18 FORN=0TO7:POKE2040+N,14:POKEV+39+N,5:NEXT:POKEV+43,7:POKEV+42,2:POKEV+39,2
19 POKE2042,195:POKEV+41,9:POKEV+40,11:POKE2041,196
20 GOSUB215
21 PRINT"■■                                        ■":POKE2040,197
22 PRINT"■ CONTROL TOWER :              CONCORDE II  ■
23 PRINT"_______________________________________■"
24 PRINT"■                                          "
25 PRINT"■          ■              ■  ■ TIME ■    "
26 PRINT"■ ■F■ ■ ■ -F : ■              ■ |______|  "
27 PRINT"■ ■L■ ■ ■    : ■              ■           "
28 PRINT"■ ■E■ ■ ■      ■              ■ ■ DISTANCE ■"
29 PRINT"■ ■L■ ■ ■ -E : ■______________■            "
30 PRINT"■                          |______|   "
31 PRINT"■                                   "
32 PRINT"■                                   "
33 PRINT"■ ■ UNDERCARRIAGE ■ |■DOWN■|            "
34 PRINT"■                                   "
35 PRINT"■ ■ SPEED ■|       |■ HORIZON ■   ■ MAP ■  "
36 PRINT"■                                   "
37 PRINT"■                                   "
38 PRINT"■ ■ FLAPS ■|■ UP ■|                  "
39 PRINT"■                                   "
40 PRINT"■ ■ ALTITUDE ■[       ]            "
41 PRINT"■_______________________________________"
42 PRINT"] 9:8:7:6:5:4:3:2:1:0:1:2:3:4:5:6:7:8:9  "
43 PRINT"■ WEST________■DEVIATION■________EAST ":POKEV+21,255
44 DATA255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
45 DATA255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
46 DATA255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
47 DATA255,255,255,255,255,255,255,255,255
48 REMTL
49 DATA248,0,0,255,192,0,255,254,0,255,255,240,255,255,255,255
50 DATA255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
51 DATA255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
52 DATA255,255,255,255,255,255,255,255,255,255,255
53 REMTR
54 DATA0,0,31,0,3,255,0,127,255,15,255,255,255,255,255,255
55 DATA255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
56 DATA255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
57 DATA255,255,255,255,255,255,255,255,255,255,255
58 REM BR
59 DATA0,68,0,0,143,192,0,143,128,0,31,0,1,79,224,1,31,192,0,47,0,4,31,128,30
60 DATA63,128,62,47,192,124,7,192,248,71,192,248,31,224,248,127,216,112,191,248
61 DATA0,63,252,0,127,252,1,251,240,0,7,252,9,255,240,7,192,0
62 REM LARGER
63 DATA0,60,0,0,126,0,0,255,0,1,255,128,3,255,192,7,255,224,15,255,240,31,255
64 DATA248,63,255,252,127,255,254,255,255,255,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
65 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0
66 REM AVA
67 DATA0,60,0,0,231,0,3,129,192,14,0,112,56,24,28,224,24,7,224,24,7,56,24,28
68 DATA14,4,112,3,129,192,0,231,0,0,60,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
69 DATA0,0,0,0,0,0,0,0
70 REM SMALLR
71 DATA0,60,0,0,126,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
72 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
73 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0
74 REMMIDDLER
75 DATA0,60,0,0,126,0,0,255,0,1,255,128,3,255,192,0,0,0,0,0,0,0
76 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
77 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0
78 A=PEEK(197):AA=NOTPEEK(56320)AND15:F=PEEK(56320)AND16:P=INT(RND(1)*4)
79 IFAL<20ORDI>380THEN84
80 IF(AAAND4)ANDTT=2THENFORN=0TO3:POKE2044+N,13:NEXT:TT=1:Y1=Y1-9:Y3=Y3+9
81 IF(AAAND8)ANDTT=2THENFORN=0TO3:POKE2044+N,15:NEXT:TT=3:Y1=Y1+9:Y3=Y3-9
82 IF(AAAND4)ANDTT=3THENFORN=0TO3:POKE2044+N,14:NEXT:TT=2:Y1=Y2:Y3=Y2
83 IF(AAAND8)ANDTT=1THENFORN=0TO3:POKE2044+N,14:NEXT:TT=2:Y1=Y2:Y3=Y2
84 POKEV+10,X1:POKEV+11,Y1:POKEV+12,X2:POKEV+13,Y2:POKEV+14,X3:POKEV+15,Y3
85 POKEV+8,HX:POKEV+9,HY:POKEV+6,FX:POKEV+7,FY
86 IFTA=1THENGOSUB146
87 IFTA=2THENPOKEV+2,INT(TX):POKEV+3,INT(TY):GOSUB191
88 POKEWA,0:POKEWA,129:POKEHI,HH:POKELO,HH+10:POKEAT,190
89 POKEW3,0:POKEW3,129:POKEH3,HH1:POKEL3,HH1+10:POKEA3,190
90 DI=DI+(S/1000):FY=FY+(S/15000):HH=INT(S/30):HH1=INT(S/12)
91 IFDI>DTTHENDT=DI+17:BY=BY+1
92 POKEV+4,INT(BX):POKEV+5,INT(BY):POKEV,AX:POKEV+1,AY
93 IFDI>150ANDDI<153THENBX=BX+1
94 IFS<100ANDTA=1ANDAL<40THEN99
95 IFTT=1ANDAX<250THENAX=AX+4:TY=TY+1
96 IFTT=3ANDAX>70THENAX=AX-4:TY=TY+1
97 IF(AAAND2)ANDS>100THENAL=AL+15:IFY1<131THENY1=Y1+3:Y2=Y2+3:Y3=Y3+3:HY=HY+1
98 IF(AAAND1)ANDAL>0THENAL=AL-5:IFY1>88ANDY3>88THENY1=Y1-3:Y2=Y2-3:Y3=Y3-3:HY=HY
-1
99 IFF=0ANDS<2000THENS=S+20
100 IFS<50ANDAL>20THENAL=AL-2
101 IFA=13ANDS>0THENS=S-20
102 IFDI>SSANDDI<SETHENGOSUB153
103 IFDI>378.7ANDDI<380THENTA=2:POKE2041,198:TY=Y2
104 IFDI>375ANDDI<377THENA$="RUNWAY APPROCHING
105 IF(AAAND2)THENTY=TY+3.3
106 IF(AAAND1)THENTY=TY-3.3
107 IFA=10ANDAL>20THENAL=AL-20
108 PRINT"■■■■■■■■"TAB(32)RIGHT$(TI$,4)
109 PRINTT$"■■■■■■■■■■■"S"■ ":PRINTT$"■■■■■■■■■■■■■■■■■■■■■■"AL"■ "
110 IFA=3THENPRINTT$"■■■■■■■■■■■■■■■■DOWN■":FL=1
111 IFA=6THENPRINTT$"■■■■■■■■■■■■■■■■ UP ■":FL=0
112 IFA=5THENPRINTT$"■■■■■■■■■■■■■■■■■■■■■■DOWN■":UC=1
113 IFA=4THENPRINTT$"■■■■■■■■■■■■■■■■■■■■■■ UP ■":UC=0
114 PRINT"■■■■■■■■■■■":PRINT"■"TAB(32)INT(DI):IFY2>120ANDS>0THENS=S-2
115 IFY1<100ANDAL>0THENAL=AL-2
116 IFY2<95ANDS>0THENS=S+10
117 IFY1>120THENAL=AL+10
118 IFS>300ANDUC=1THENA$="TOO FAST : UNDERCARRIAGE DOWN ; DAMAGED":GOTO138
119 IFS>200ANDFL=1THENA$="TOO FAST : FLAPS DOWN ; DAMAGED":GOTO138
120 IFS>100ANDFL=0ANDS<240THENA$="NO FLAPS":GOTO138
121 IFAL<300ANDUC=0THENA$="YOUR UNDERCARRIAGE UNDER 300 FT":GOTO138
122 IFAL<5ANDTA=0THENA$="YOU HAVE CRASHED INTO THE GROUND":GOTO138
123 IFS<100ANDTA=0THENA$="YOU HAVE STALLED : TOO SLOW":GOTO138
124 IFS<10ANDTA=2THENA$="STALLED : MUCH TO SLOW":GOTO138
125 IFDI>383THENA$="YOU HAVE FLEW PASS THE AIRPORT":GOTO138
126 IFS>1000ANDDI>250THENA$="YOU ARE TRAVELLING TOO FAST:■SLOW DOWN■■"
127 IFS>500ANDDI>300THENA$="APPROCHING AIRPORT: ■SLOW DOWN■■
128 IFFY>125ANDFY<129THENA$="■■ LOW ON FUEL ■
129 IFFY>130THENAL=AL-3:S=S+2
130 PRINT"■■■■ "A$
131 IFP>1THENA$="
132 IFP<1THENA$="YOU'RE DOING FINE
133 IFAX<80ORAX>240THENA$="YOU ARE OFF COURSE
134 IFAL<300ANDTA=0THENA$="YOU ARE LOW
135 IFTA=1THENA$="CLEAR FOR TAKE OFF : SPEED 100 MPH
136 GOTO78
137 REM DEATH
138 POKEV+21,0:PRINT"■■":POKEV+33,0:POKEV+32,0:PRINTA$"!"
139 FORN=1TO3000:NEXT
140 SC=TI/50+DI:IFSC>HSTHENHS=SC
141 PRINT"■■■SCORE : "INT(SC)"    HIGH SCORE :"INT(HS)
142 A=PEEK(197):IFA=64THEN142
143 DI=0:FL=0:UC=1:S=0:AL=0:SC=0:HH=0:HH1=0
144 GOTO13
145 REM TAKE OFF
```



```
146 IFDI>1ANDAL=0THENA$="RAN OUT OFF RUNWAY":GOTO138
147 IFDI<1THENTX=X2:TY=Y2
148 IFDI>1.1THENTA=0
149 POKEV+2,TX:POKEV+3,TY
150 IFDI>1.2ORDI<1.6THENTX=5:TY=5
151 RETURN
152 REM STORM (?)
153 IFDI>SS-20ANDDI<SSTHENA$="WEATHER : WINDS 200MPH : VISIBILITY 0  "
154 LI=INT(RND(1)*15)
155 IFLI>1ORDI<SS+23ORDI>SE-23THEN157
156 IFSG=0THENPOKEV+33,1:SG=1:FORN=0TO2:POKEV+44+N,5:NEXT:GOTO158
157 IFSG=1THENPOKEV+33,11:SG=0:FORN=0TO2:POKEV+44+N,11:NEXT
158 IFDI>SS+3ANDDI<SS+7THENPOKEV+33,6:POKEV+40,7
159 IFDI>SS+7ANDDI<SS+12THENPOKEV+33,15:FORN=0TO2:POKEV+44+N,11:NEXTN
160 IFDI>SE-18ANDDI<SE-12THENPOKEV+33,12
161 IFDI>SE-12ANDDI<SE-7THENPOKEV+33,15:FORN=0TO2:POKEV+44+N,11:NEXTN
162 IFDI>SE-7ANDDI<SE-3THENPOKEV+33,6:FORN=0TO2:POKEV+44+N,5:NEXT:POKEV+39,2
163 IFDI>SE-4THENPOKEV+33,14:POKEV+40,11
164 IFDI>SS+12ANDDI<SS+17THENPOKEV+33,12:POKEV+41,3:POKEV+39,3
165 IFDI>SS+18ANDDI<SS+23THENPOKEV+33,11
166 IFLE=1THENRETURN
167 REM STORM 1
168 IFLI=2ANDAX<245THENAX=AX+6
169 IFLI=3ANDAX>20THENAX=AX-6
170 IFLE=2THENRETURN
171 REM STORM 2
172 IFLI=4ANDY2>90ANDTT=2THENY2=Y2-3:Y1=Y2:Y3=Y2:HY=HY-1
173 IFLI=5ANDY2<130ANDTT=2THENY2=Y2+3:Y1=Y2:Y3=Y2:HY=HY+1
174 IFLE=3THENRETURN
175 REM STORM 3
176 IFLI=6ANDAL>100THENAL=AL-10
177 IFLI=7THENS=S-10
178 IFLE=4THENRETURN
179 REM STORM 4
180 IFLE=8THENAL=AL+20
181 IFLE=9ANDTT=2THENY1=Y1-9:Y3=Y3+9:TT=1:FORN=0TO3:POKE2044+N,13:NEXT
182 IFLE=10ANDTT=2THENY1=Y1+9:Y3=Y3-9:TT=3:FORN=0TO3:POKE2044+N,15:NEXT
183 IFLE=4THENRETURN
184 REM STORM 5
185 IFLI=11THENDI=DI+10
186 IFLI=12THENFY=FY+.1
187 IFLI=13ANDAL>150THENAL=AL-100
188 IFLI=15ANDAX<>20THENAX=AX-20
189 RETURN
190 REM LANDING
191 TX=AX-8:IFTX<>240THENTX=Y3
192 IFTX<72THENTX=Y1
193 IFTY>Y376ANDTT=2THENTY=Y1
194 IFTY>Y1ANDTT=2THENTY=Y2
195 IFTY>Y2ANDTT=2THENTY=Y3
196 IFDI>=300ANDDI=<300.3THENPOKE2041,199
197 IFDI>300.3THENPOKE2041,196
198 IFTY<107ANDPEEK(2041)=196THENTY=107
199 IFS>150THENA$="SLOW DOWN                             ":GOTO201
200 IFAL>150THENA$="YOU'RE MUCH TOO HIGH                  ":GOTO201
201 POKEV+2,INT(TX):POKEV+3,INT(TY)
202 IFDI>300.8THEN207
203 RETURN
```

```
204 A=PEEK(197):IFA=64THEN204
205 PRINTA:GOTO204
206 REM LAND ?
207 IFAL>500RS>100THENA$="YOU PASSED RUNWAY":GOTO138
208 IFTT=1ANDAL=<20THENA$="YOUR LEFT WING HIT THE RUNWAY":GOTO138
209 IFTT=3ANDAL=<20THENA$="YOUR RIGHT WING HIT THE RUNWAY":GOTO138
210 IFAX=<160ORAX>=190THENA$="YOU HIT THE GRASS":GOTO138
211 IF(AL>=20ANDAL=<50)AND(S=<100ANDS>=40)THENA$="ROUGH LANDING":GOTO138
212 IFAL<=20ANDS=<40ANDS>=30THENA$="JUST MADE IT":GOTO138
213 IFAL<=10ANDS<30THENA$="NICE LANDING":GOTO138
214 REM LEVELS
215 POKE53281,0:POKE53280,0:PRINT"▒"
216 PRINT"  ▒▒▒▒ ▒▒▒▒ ▒▒  ▒ ▒▒▒▒ ▒▒▒▒ ▒▒▒▒ ▒▒▒▒ ▒▒▒▒"
217 PRINT"  ▒   ▒ ▒ ▒▒▒  ▒ ▒  ▒ ▒ ▒ ▒  ▒ ▒ ▒"
218 PRINT"  ▒   ▒ ▒ ▒ ▒▒ ▒ ▒  ▒ ▒ ▒▒▒  ▒ ▒ ▒▒▒"
219 PRINT"  ▒   ▒ ▒ ▒  ▒▒ ▒ ▒  ▒ ▒ ▒ ▒  ▒ ▒ ▒"
220 PRINT"  ▒▒▒▒ ▒▒▒▒ ▒  ▒▒ ▒▒▒▒ ▒▒▒▒ ▒ ▒ ▒▒▒▒ ▒▒▒▒▒▒"
221 PRINTTAB(13)"▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
222 PRINTTAB(13)" ▒▒▒▒▒▒▒▒▒▒"
223 PRINTTAB(13)"   ▒▒  ▒▒"
224 PRINTTAB(13)"   ▒▒  ▒▒"
225 PRINTTAB(13)"   ▒▒  ▒▒"
226 PRINTTAB(13)"   ▒▒  ▒▒"
227 PRINTTAB(13)"   ▒▒  ▒▒"
228 PRINTTAB(13)"   ▒▒  ▒▒"
229 PRINTTAB(13)"   ▒▒  ▒▒"
230 PRINTTAB(13)"   ▒▒  ▒▒"
231 PRINTTAB(13)" ▒▒▒▒▒▒▒▒▒▒"
232 PRINTTAB(13)"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
233 PRINTTAB(12)"BY WILLIAM FONG"
234 FORN=10TO240:POKEWA,0:POKEWA,33:POKEHI,N:POKELO,N+10:POKEAT,56
235 A=PEEK(197):IFA<>64THEN237
236 POKEW3,0:POKEW3,129:POKEH3,N-10:POKEL3,N:POKEA3,56:NEXT:GOTO234
237 PRINT"▒▒  CONCORDE II : WILLIAM FONG  (C)1984 ▒▒▒▒"
238 PRINTTAB(4)"▒▒    ▒▒▒▒ ▒   ▒ ▒▒▒▒ ▒▒    ▒▒▒▒▒"
239 PRINTTAB(4)"▒▒    ▒▒   ▒  ▒  ▒▒   ▒▒    ▒▒  "
240 PRINTTAB(4)"▒▒    ▒▒▒  ▒ ▒▒  ▒▒▒  ▒▒    ▒▒▒▒ "
241 PRINTTAB(4)"▒▒▒▒  ▒▒   ▒▒    ▒   ▒▒▒▒   ▒▒▒"
242 PRINTTAB(4)"▒▒▒▒▒▒ ▒▒▒▒  ▒   ▒▒▒▒ ▒▒▒▒▒▒ ▒▒▒▒ ▒▒▒▒▒"
243 PRINTTAB(7)"LEVEL 1 : BEGINNER▒"
244 PRINTTAB(7)"LEVEL 2 : AMATEUR▒"
245 PRINTTAB(7)"LEVEL 3 : PILOT▒"
246 PRINTTAB(7)"LEVEL 4 : EXPERIENCE PILOT▒"
247 PRINTTAB(7)"LEVEL 5 : PROFESSIONAL▒▒▒▒"
248 PRINTTAB(9)"▒ ( CHOOSE A LEVEL ) "
249 GETL$:IFL$=""THEN249
250 LE=VAL(L$):IFLE<1ORLE>5THEN249
251 IFL$="1"THENSS=0:SE=0
252 IFL$="2"THENSS=100:SE=300
253 IFL$="3"THENSS=90:SE=330
254 IFL$="4"THENSS=80:SE=350
255 IFL$="5"THENSS=70:SE=370
256 POKE53281,14:RETURN
```



Toiling deep within his murky cellar, Runecaster opens the portals to Other Worlds . . .

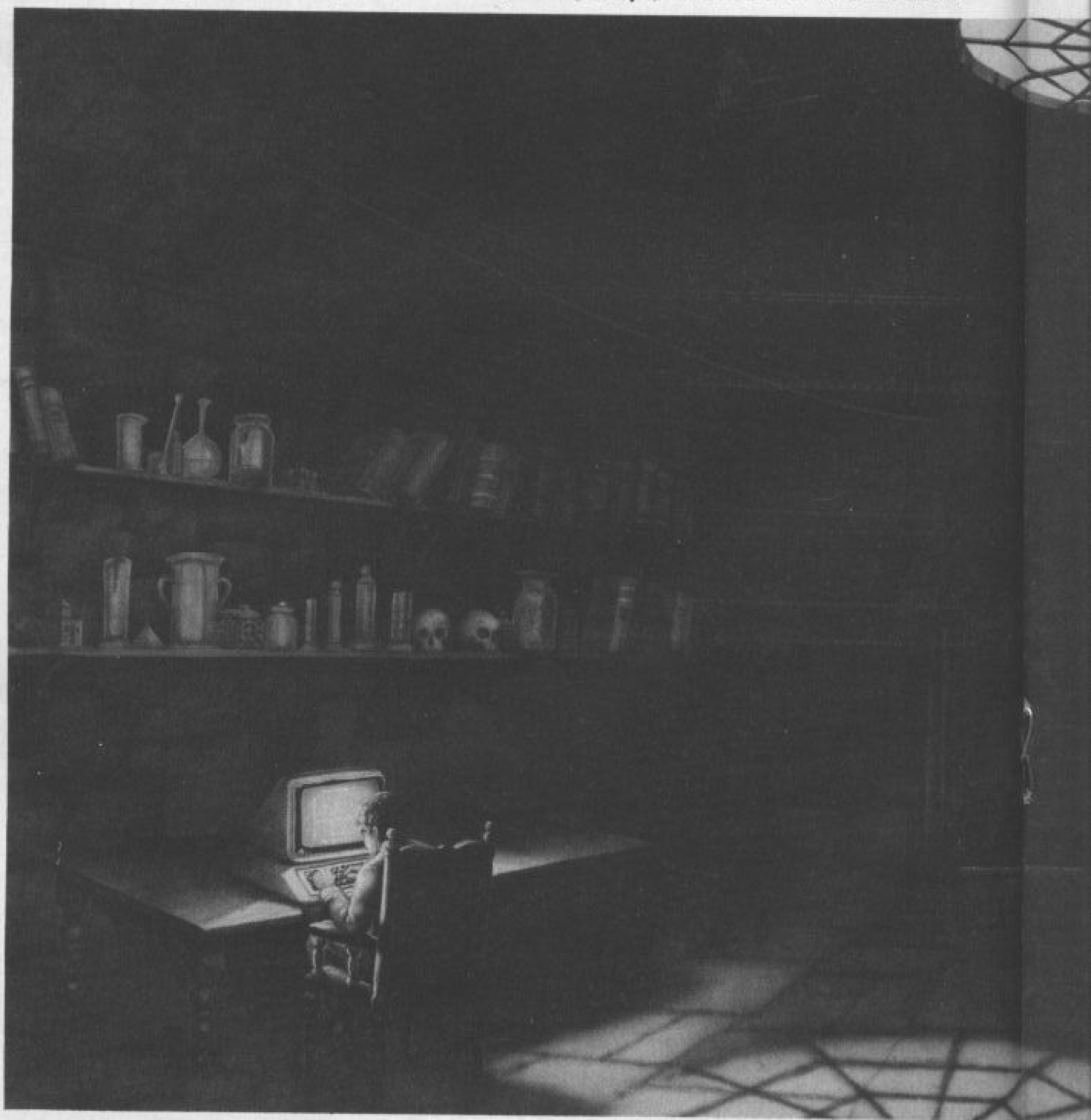
TALES FROM THE CRYPT

WHAT IS AN ADVENTURE program? Today we look on adventures as part of our everday (computer) lives. They come in all shapes and sizes for all the different computers we can buy — where did they start? Many of us today, especially those reading this, own a personal home computer. Just a few years ago they did not exist — hard to believe, isn't it? Five or six years ago, the options were severely limited — PETs, TRS-80s, build-your-own, a memory capacity standard of about 8K, some 16K or 32K perhaps, if you were rich. Those were the days when computers and computer talk were still the closely-guarded 'hi-tech' secret of just a few people who would go to almost any length to promote, not knowledge, but bafflement and confusion to those who were not 'signed on the line' members of that close-knit club.

Gone, but. . .

Fortunately those days are past, not long in time, but dead, dead, dead. The home computer has come of age — young but strong enough to walk on its own. But . . . one thing those mainframe freaks did have that we can thank them for being unable to keep to themselves was adventure, or more properly, Adventure. For, among the dim corridors of finance, medical statistics, research into this and that, was to be found (often illicitly), Adventure. The time was just right for its birth: Dungeons and Dragons was already coverting more and more to its banner — a game (or way of life) where you, ordinary mortal, could enter a world so different yet so real, that your life could depend on you knowing the right spell or your being adept at hearing the very fabric of a simple chest whispering to you that it contained a dozen poison arrows, primed and ready for action.

Into this environment came Adventure, written (I believe) by Messrs Crowther and Woods around 1978. What was it? Adventure was a new sort of game (life?) where you could enter an amazing world via your computer terminal — many of them did not even have a VDU. So that meant huddling over some sort of 'teletype' to see the answer to your simple instruction of GO SOUTH. This 'world' within the computer was like no board game where you can see quite clearly where you will 'land' if you throw a '6'. All you knew was the direction in which you could move — sometimes not even that! What was in that direction, you knew not. You had to explore your surroundings and gradually build up a map of them. I'm sure that one of the most important attractions is this freedom



to move where you choose, not bounded by the finite 'walls' imposed by the conventional game.

Back to Adventure. There were many passages to explore, logical puzzles to solve and treasure to find. Adventure caught on like no other game before or since. Research scientists and computer operators, designers and programmers spent (wasted?) hours trying to get past the Troll! Unlike many of the successful arcade games which fade into history after months, Adventure has stood the test of time. Many programmers have tried to copy or improve upon it — a few have succeeded, but only a few.

We have moved to an age that expects pretty pictures with our adventures; sometimes it is well-done and the pictures play a part in the adventure. Mostly the graphics that some reviewers rave about are only there to enable a product to be 'sold' to the public — some distributors (and much of what is in your local store is determined by the distributors) will not even take adventures unless they have those magic, 'selling' graphics!

Graphics take up memory that could otherwise be used to make the program more interesting and present you with more of a challenge — significantly setting the scene to make you feel you are really there! The written word is *very* powerful, not just by itself but in combination with your imagination. Mind you, you either have that sort of mind — or you don't! Many people (poor souls) cannot 'get into' 'Lord of the Rings' by J. R. R. Tolkien; those that can are able to populate that world with beings so complete as to be real. The same comments may be made of Larry Niven's science fiction. How many of us have been disappointed at seeing an artist's attempt to represent a scene from a book we love? And an artist has at his disposal a far, far greater resolution than even the best monitor with a BBC Model B in Mode 0 can ever hope to attain. Graphics are fine but cannot for the foreseeable future replace the written word.



Adventure *used* the written word as an author does: descriptions of places and things were full and rich, not short-form "you are in a cave NSE'. As the original was mainframe based with megabytes of core store, it did take some time before it appeared on home computers! But one way and another, with the sophistication of text compression and the dedication of the programmers, you too can own a version of the first of its type, whatever it may be called! Versions have been around for the Spectrum and Nascom for some years. Now we also have them for the BBC and of course our special interest, the CBM 64.

Two versions are produced by software houses that have a good reputation, so it is very much a question of 'you pays your money and takes your choice'. The two programs are Colossal Adventure by Level 9 Computing, and Classic Adventure from Melbourne House. Both are based heavily on the original but in the case of Colossal Adventure, the number of locations has been increased by 70 and a completely new end-game has been added! Classic Adventure has a fast LOADing system (Pavloda) which cuts LOADing time to three minutes; on the review sample this only worked one time in four, so both DO take some time to LOAD! Level 9 are hoping to have a fast LOAD version later in the year.

Both are text-only — but what a text! Level 9 seems to have someone with a really imaginative mind writing their scripts! Even so, Classic Adventure is far more descriptive than the average run of text adventures. 'Classic' follows the mainframe more accurately when it comes to the mazes. Whereas I can refer to my much-copied, faded, almost unreadable map, culled from a mainframe version and escape (cheat!) easily from 'Classic', I have to actually work at it in 'Colossal'.

'Classic' has the odd spelling mistake and a few anomalies: UNLOCK GRATE — OK: OPEN GRATE — THERE IS NOTHING HERE WITH A LOCK. This jars a little but does not spoil the overall adventure. GET 'A' AND 'B' — OK. But only 'A' is actually taken — neither game allows these multiple statements.

'Colossal' has screen colours that do not entirely suit my television but nevertheless are quite readable.

Choose one. . .

Whichever you choose, do get one of them. So much has been written over the years about Adventure that although you may not realise it, you will have subconsciously picked up some threads of the plot. When you start playing you may well occasionally get the feeling that you have been here before.

Adventure games give you the opportunity of exploring another world, place or time. Part of their attraction is this feeling of freedom of decision amplified by our own imaginations. It is said that one picture is worth a thousand words, but when that one picture appears within the framework of an adventure game, it is my opinion that it destroys the one thing that adventure games give me — the total freedom for my imagination to run wild!

INTRODUCING MODEMS

Modems seem to be the 'in thing' these days for computer freaks, but how many people know what they can do? Simon Rockman introduces them and their uses to Commodore owners.

A MODEM IS ONE OF THE most exciting black boxes that can be hooked up to a computer. Think of it as a telephone for your computer. The device plugs into the back of a computer using either the edge connector or the cartridge port. On the back of the modem is a lead with the new style telephone jack. This plugs into the BT approved hole in the wall just like a Mickey Mouse telephone. From then on the world is your oyster, but as with a telephone it is no use if you've got no-one to ring. There are quite a few major services to dial into.

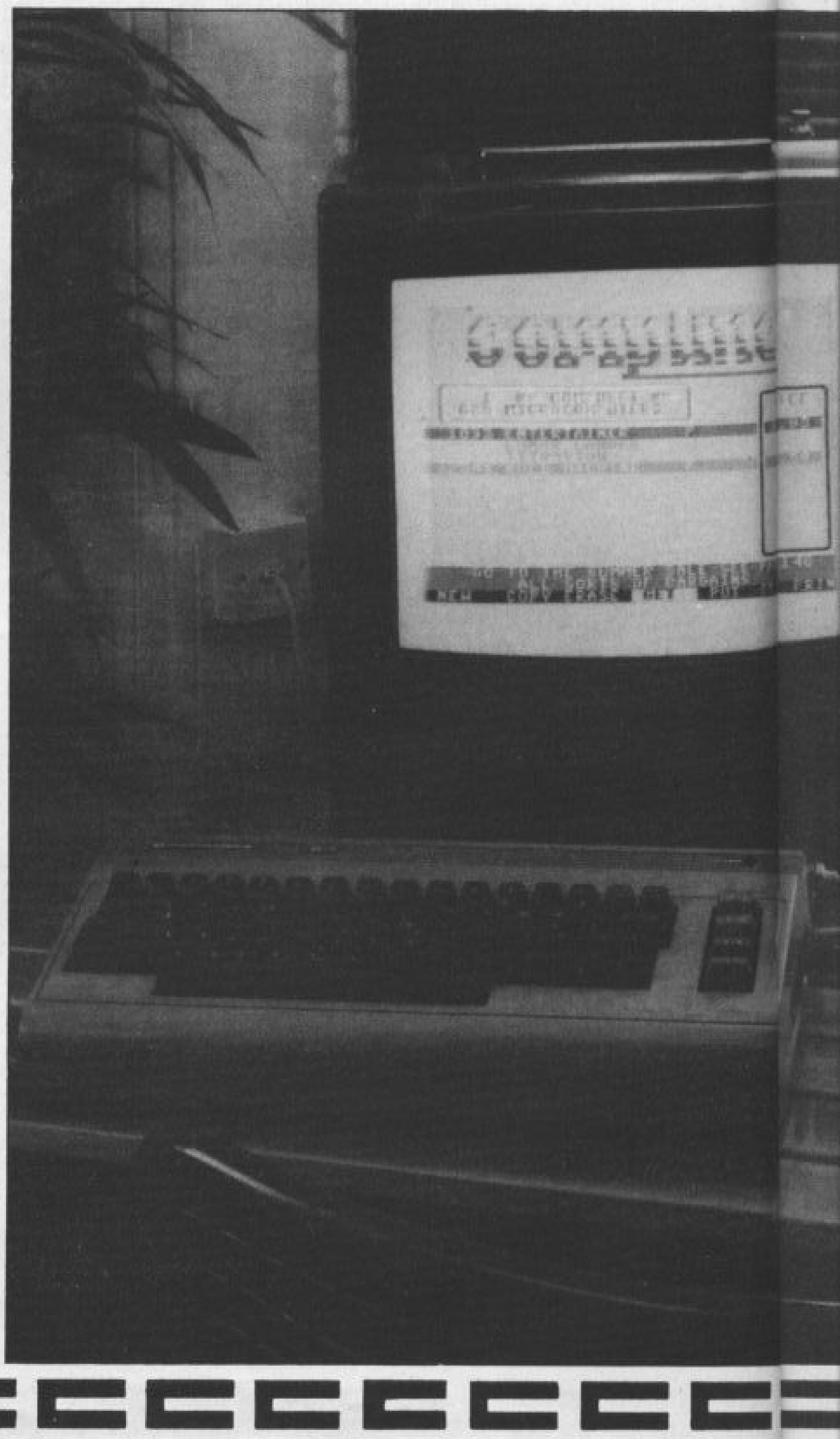
Main contenders

The major database used by home computer owners is PRESTEL. This covers the whole country, but most people only have to make a local call to log in. To use PRESTEL you have to pay a standing charge of at least five pounds a quarter and this allows you to log into the general areas of the system. You are given a ten digit ID and a four digit password, the ten digits are fixed and can only be changed by PRESTEL themselves, the second one is like a combination lock and can be changed by the user. Both of these numbers should be kept secret and for added security the second one should be changed regularly. Prestel provides most of the information found in a general magazine — news, sports and holiday information on a specific topic than the system provides it may be possible to join a Closed User Group (CUG). These usually cost extra to join and are particularly popular with travel agents, farmers and micro owners. The section aimed at micro owners is called "PRESTEL Microcomputing" and incorporates the association of computer clubs, Viewfax and Micronet.

Micronet

Micronet is run like a daily newspaper, it is kept up to date with news daily, often being the best way to keep track of what is happening in the world of computers (apart from buying magazines produced by yours truly! Ed.) There are special technical enquiry facilities and programs to download, some of them quite good, such as "The Hobbit". At the moment there is very little specifically for the Commodore 64 user. However, with the increasing number of 64 modems available, the force of Commodore owners on Prestel should soon begin to challenge the Spectrum and BBC strongholds. The Micronet letters page is very addictive; Prestel is a two-way system and this makes the most of that. The micromouse section of Micronet is a kind of gossip column, spouting rumors and statistics. Viewfax is a less formal version of Micronet offering many similar services but with a flavour of its own. Access to Prestel Microcomputing costs an extra eight pounds a quarter. One way an information provider (IP) can charge you money for information is by putting a price on a page. When you look at the charged page the amount in the top right hand corner is added to your bill. This is the way a lot of IP's make their money.

On the whole, Prestel Microcomputing can be used very cheaply — my average bill for Prestel, including the Microcomputing section plus the odd



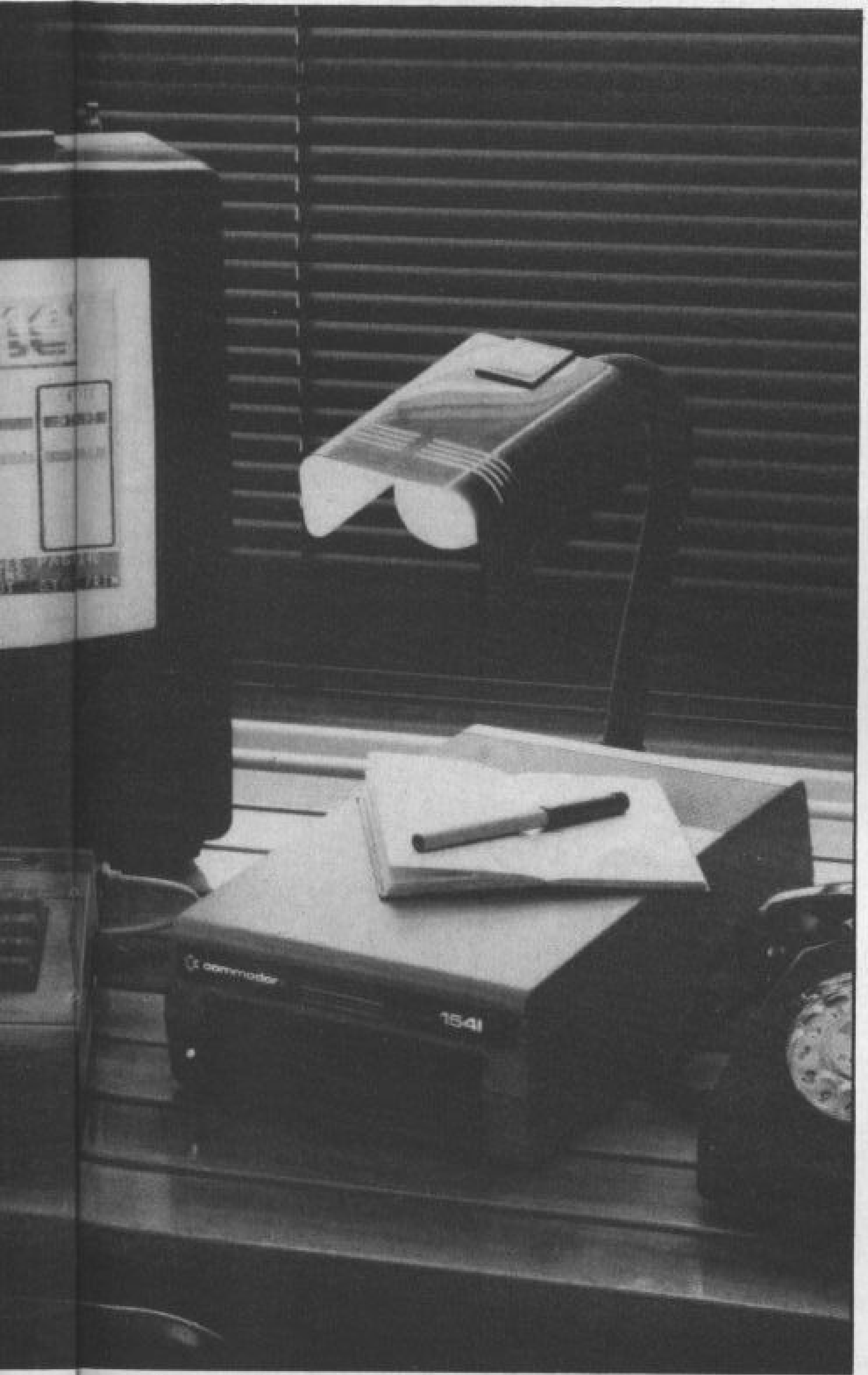
page charge is around £15 a quarter, about the same sort of price as a daily newspaper.

Into business

There are many business orientated networks; most of them require an eighty column display and so are not really suitable. They also tend to be rather expensive. The main system, which will be available to Commodore 64 owners is British Telecom Gold. This rose to fame when it was broke into on the BBC television programme, but claims to be much safer now. Gold is a messaging system: it can be used to send long letters and programs to other users, you can store as much information as you like but pay for the amount of memory this takes up. The system is only really useful if you log in regularly or want to send information that is difficult to read over the phone, like legal documents or programs.

Latest bulletin

You don't have to have a huge mainframe to run a database, there are many micros doing a similar job. These are called bulletin boards (BBs). They usually run on a Tandy computer and are free. Most BBs use a slow rate of data transfer and are not compatible with the system used by Prestel; this means that only modems which are capable of changing the speed at which they operate (known as a Baud rate) can 'talk' to these systems. With the increasing popularity of Prestel more and more BBs are switching to the system that Prestel uses. The advantage of a BB is that whatever you send in is put up instantly, everyone is an information provider. The disadvantage is that only one person can use the system at a time. The popular BBs can be very difficult to get through to, they are usually engaged. BBs are only local and it is necessary to dial the site where the computer is set up.



Getting switched on

Bigger computers usually go through a switching system; the main one is PSS. This covers most of the country and is a kind of motorway for data: you dial a local number on the grid (a node), enter an identification number, a password and who you want to talk to and the system puts you through. This service is pretty cheap, data is charged for per packet (a number of bytes) and is much more economical than calling direct long distance, especially when dealing with other countries. PSS requires the person you want to talk to to also be on the system.

Compunet

The most exciting database from the Commodore 64 owner's point of view is 'Compunet' specially set up by Commodore in conjunctin with a company called ADP. To use Compunet it is necessary to buy Commodore's own modem.

Commodore have been spectacularly successful in the US with their two modems for the VIC and 64. They hope to repeat this in Europe by turning the telephone into the computer's most useful peripheral. There are major differences between the American standards for modems and European ones so it was decided to build a new one from scratch in the UK. The job of designing the hardware and writing the Viewdata software was given to Y2 in Watford. This has been around for a fair while now and has an established user base. With the modem comes special software to allow you to use Compunet.

When you log into Compunet for the first time all the extra software that you need is sent to you by the system. This includes a routine to allow you to talk to other users direct. Compunet is designed as a cross between Prestel and a BB. All users are information providers but they can charge for the information they provide. There is a system of menus and you have the ability to communicate directly with other users and there are plans for a multi-player game like the American "Mega wars" or Essex "MUD" space and adventure games which allow you to play against other players in real time. Each Commodore modem is unique; it contains a code number (like the registration on a car) which tells the system who is calling. This adds a great amount of security to the system and it is hoped that holidays, full home-banking and betting will be sold through Compunet.

Commodore have built 3000 modems, they have the parts for 7000 and aim to sell 40,000 by Christmas. One person I spoke to said "We will make Micronet look silly". These plans are certainly ambitious, other computers will be allowed in after a while (Apple are known to be interested), but it is the Commodore 64 owner who will get first crack of the whip. Watch future issues of Your Commodore for details.

MAKING MUSIC

Take those earplugs out and flex those fingers — William Fong has written a helpful music program for those learning to play a musical instruement.

MAKING MUSIC ON THE Commodore 64 using Commodore BASIC is a long and tiring process. I have written a program which will allow you to play any tune you wish and have it played back at your chosen speed and instrument.

When you RUN the program the screen will display the treble clef and a summary of the instructions. You do not play the notes as you would on a piano but you have to press the notes' names, and when you do the notes will be dispayed on the scales in their correct places. I have done it this way because you will not be learning if you pressed any old note; I had to go through this stage when learning the piano. For those who do not play any instruments here is a summary of the notes playable:

MIDDLE C-D-E-F-G-A-B-C-D-E

Instructions

(+) This will play back the music you have just created.
(CBM + PLAY KEY) This will play the sharp of the note depressed.
(←) Stops music from continuing play back.
(CLR/HOME) This will clear all the variables giving you a fresh start.
(-) This will delete a note.
(1-9) The numbers from one to nine will play the playback at different speeds, one being the fastest.
(★) This will play a one beat rest and display a rest on the scale.
(↑) This will display all the data needed to produce the tune you have created. This is very useful as writing music for your own program is speeded up.
(FUNCTION KEYS) Theses will choose the different instruments.
(SHIFT WITH NOTE) This will give you a note an oclave higher. N.B. The highest note is high E

This is only a very short program for what it does but it will be very helpful for the beginner who's learning to play any instrument.
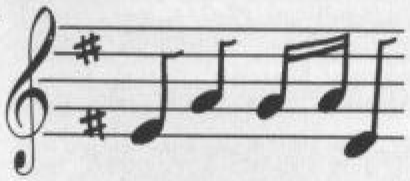
When the whole scale is full



PROGRAM SPECIALS

1-9	SETTING UP VARIABLES
10-16	DISPLAY SCREEN
20-180	GET ROUTINE
190-220	PLAY NOTE AND PRINTS CROTCHET ON SCALE
230-250	DISPLAY DATA FOR TUNE
300-320	PLAYBACK ROUTINE

the scale will be cleared; this does not mean you have lost all of the tune: it is still stored in memory and you can have up to four hundred notes. By changing the DIM statement near the start you can have more notes. You can play endless tunes and make your programs much more enjoyable: it depends on you.



Program Listing

```
1 CLR:POKE54296,10:POKE53281,8:POKE53280,8:PL=54274:PH=54275:O=100:W=65:AD=9
5 L$="■┼────────┼────────┼────────┼────────┼■":Q$="■"
9 HI=54273:LO=54272:A=54277:WA=54276:DIML(400),H(400)
10 Z=0:PRINT"■MUSIC MASTER  ■■ BY WILLIAM FONG (C) ■■■":FORN=1TO5:PRINTL$:NEXT
12 PRINT"■■■ ■[+]■ PLAY MUSIC      ■[-]■ DELETE NOTE           ■[*]■ REST"
14 PRINT" ■[CBM FLAG]■ SHARP   ■[SHIFT]■ OCTAVE HIGHER ■[←]■ STOP PLAY"
15 PRINT" ■[CLR/HOME]■ FRESH START":PRINT" ■■[F1] ■PIANO":PRINT"■ ■[F3] ■FLUTE"
16 PRINT" ■■[F5] ■FANTASY":PRINT" ■■[F7] ■VIOLIN■          ■■[1-9]■ SPEED OF PLAY"
19 N$="■●■ ■ ■ |"
20 GETA$:IFA$=""THEN20
23 IFA$="↑"THEN230
25 IFA$="¯"THENM=1:H=43:L=52:GOTO190
27 IFA$="−"THENM=2:H=38:L=126:GOTO190
30 IFA$="B"THENM=4:H=32:L=94:GOTO190
35 IFA$="−"THENM=3:H=34:L=75:GOTO190
40 IFA$="A"THENM=5:H=28:L=214:GOTO190
45 IFA$="G"THENM=6:H=25:L=177:GOTO190
50 IFA$="F"THENM=7:H=22:L=227:GOTO190
55 IFA$="E"THENM=8:H=21:L=154:GOTO190
60 IFA$="D"THENM=9:H=19:L=63:GOTO190
65 IFA$="C"THENM=10:H=17:L=37:GOTO190
80 IFA$="■"THEN1
85 IFA$="+"THENGOSUB300
90 IFA$="-"THENPRINT"■";TAB(Z)"■■ ■■ ■■├■■ ■■├■■ ■■├■■ ■■├■■ ■■├■■ ■■ ■■ ":Z=Z-2
:P=P-1
95 IFP=<0THENP=0
96 IFZ<=0THENZ=0
98 IFA$="*"THENU=2:N$="■■/■■\■■/■■\":L=0:H=0:GOTO190
100 IFA$=" ┌"THENM=5:H=30:L=141:U=1:GOTO190
115 IFA$=" ▀"THENM=3:H=36:L=85:U=1:GOTO190
120 IFA$=" ▪"THENM=9:H=20:L=100:U=1:GOTO190
125 IFA$="▪ "THENM=7:H=24:L=63:U=1:GOTO190
130 IFA$="| "THENM=6:H=27:L=56:U=1:GOTO190
135 IFA$=" ┌"THENM=5:H=30:L=141:U=1:GOTO190
140 IFA$="■"THENW=65:AD=9
145 IFA$="■"THENW=17:AD=167
150 IFA$="■"THENW=33:AD=15
155 IFA$="■"THENW=65:AD=167
160 IFA$="1"THENO=30
162 IFA$="2"THENO=60
164 IFA$="3"THENO=100
166 IFA$="4"THENO=150
168 IFA$="5"THENO=240
170 IFA$="6"THENO=360
172 IFA$="7"THENO=420
174 IFA$="8"THENO=480
176 IFA$="9"THENO=540
180 GOTO20
190 FORN=1TOM:X$=X$+Q$:NEXT:IFZ>35THENX$="":GOTO10
193 IFU=1THENN$="■#●■ ■ ■ |"
198 Z=Z+2:PRINT"■■■":PRINTX$;TAB(Z);N$
200 POKEWA,0:POKEWA,W:POKEPL,255:POKEPH,4:POKEHI,H:POKELO,L:POKEA,AD:P=P+1::H(P)
=H:L(P)=L
220 X$="":U=0:N$="■●■ ■ ■ |":GOTO19
230 PRINT"■■HIGH■:LOW■■":FORN=1TOP:PRINT"■"H(N)"■ ■"L(N);:NEXTN
240 GETA$:IFA$=""THEN240
250 GOTO10
300 FORN=1TOP:POKEWA,0:POKEWA,W:POKEPL,255:POKEPH,4
310 GETA$:IFA$="←"THEN20
320 POKEHI,H(N):POKELO,L(N):POKEA,AD:FORK=1TOO:NEXTK,N:RETURN
```

INPUT

Having problems? Want to moan? Like to praise? This is your chance to put pen to paper and air your views or state your case. Just write to us and see if we can help.

Dear Sir,
I am a beginner to computers, having bought a Commodore 64, with which I have no complaints. However, I do have a problem with programming. The problem is to do with the symbols which are contained in so many programs for the 64. I find it hard to distinguish the symbols from the printed sheet and I have very little understanding of these symbols and their exact meaning. I would be grateful if you could give me some tips on making the symbols clearer and also the titles of any books which will explain the symbols a little more clearly than the Commodore 64 Handbook.
Yours faithfully,
David Forrest
Renfrewshire

We answer,
The symbols can be confusing, especially when getting confused with ASCII codes, screen codes and BASIC codes. To help with this, the best book to buy would be the Programmer's Reference Guide which will help you with all aspects of programming on the CBM 64. The only other method is to persevere and deliberately use the symbols in your programs so that you will forcibly learn them and understand them.

Dear Sir,
I do not understand how to use machine code on my VIC 20. Please could you show me a small machine code program where I can hit Return after every line and type RUN at the end. In books I see long lists of letters, symbols, numbers and spaces which, when entered, give the message SYNTAX ERROR.
Yours faithfully
Samir Okasha
Cambridge

We answer,
Here is a short machine code program with a Basic loader:

```
10 p = 828
20 read a : if a=-1 then 50
30 poke p,a:p=p+1:goto 20
50 sys 828 : end
100 data 160 , 0 , 162 , 0 , 254 ,
0
120 data 30 :rem change this
to 16 for expanded vic
140 data 254, 0
160 data 31 :rem change this
to 17 for expanded vic
180 data 254 , 0
200 data 150 :rem change
this to 148 for expanded vic
220 data 254, 0
240 data 151 :rem change
this to 149 for expanded vic
260 data 232 , 208, 241 , 200 ,
208 , 238, 96 , -1
```

Dear Sir,
I have a Commodore VIC 20 and would like to expand it to its maximum. I already have two 16K and a 3K RAM Packs and a Stack 4 slot switchable mother board.
1) How can I use the second 16K RAM Pack to expand beyond the 19K that I get with one RAM Pack in place, 2) Do you think that since there are few if any programs for a 32K VIC that I am wasting my time and money in buying extras, and 3) Should I sell one of the 16K RAM Packs and buy a 32K RAM?
Yours faithfully,
A Ruff
Kingston on Thames

We answer,
Do not buy the 32K ram pack as you do not need it and also because plug-into extras like adventure cartridges, machine code monitors and so on cannot be used with it. To make full use of your packs, open them up and inside will be banks of switches or solder blocks depending on how old they are. The pins are numbered from 1 - 22. On one pack, make pins 10 and 11 and on the other pack make pins 12 and 13.

Dear Sir,
I have a VIC 20 on which I have been writing programs since last June; in the program I have written, I usually use user-defined graphics. Now I have a problem! Last Christmas, I have received a 16K RAM pack which now gives me 19.5K of usable memory. The problem is that I can't use my user-defined graphics in the way that I used to.
eg,

```
10 POKE 52,29:POKE 56,29
20 FORT=7432 TO 7463:
READ A:POKEt,A:NEXT
30 POKE 36869,255
40 DATA etc. . .
```

which defines four characters starting at screen code 33.

On the paper which comes with the RAM pack it states that "it is not possible to move the VIC screen or character set into external memory". Can you help me by publishing a program which allows you to design a character set on the internal memory with a 16K RAM Pack fitted and putting the UDGs on the screen?
Yours faithfully,
Wayne Beauchamp
Norfolk

We answer,
If you type the following:

```
10 poke 648,30 : poke 43,1 :
poke 44,32
20 poke 55,0 : poke 56,96 :
poke 2★4096,0
30 printchr$(147) : new
```

This means that your program will now work and all your existing graphic software will also work.

Dear Sir,
I have just purchased a CBM 64 which I spend many happy hours with. However I could not get my hands on a CBM tape recorder, so I bought an Altai tape deck. This works fine on loading programs which I have typed into the computer, but on some software which I have bought such as all Interceptor software and Mastertronics' Vegas Jackpot do not run. On screen it reads OUT OF MEMORY or ERROR in 10.
Is it possible to correct this fault in the tape deck or have I wasted my money?
Yours faithfully
John Adams
Belfast

We answer,
The most likely problem with this is that your recorder has no pause facility. When the computer has finished loading from casette, it automatically pauses it. For protection

OUTPUT

purposes, a lot of software houses store their products in several parts on the tape thus when loading the game, it loads one part then another. In the meantime though it has expected the tape recorder to have stopped and if it has not stopped then it has probably continued on past the next program header (or what ever start marker these companies use on their tapes — more protection). So only part of the program will load.

Dear Sir,
I want to ask you if the VIC 20 games are usable on the CBM 64; can you send me a list of the games that are usable? I have read that some people were going to expand their VIC 20 up to 32K, but I have also read in the Beginners' Handbook that the VIC 20 is 5K and expandable to 29K not to 32K! Can you clarify the situation?
Yours faithfully,
Alex Koon
London N7

We answer,
No VIC 20 games other than simple non-graphic games will run on the CBM 64. You can expand your VIC up to 38K with 23.5K available from Basic. You cannot expand the CBM 64 and this machine has 16K of ROM.

Dear Sir,
Having owned a CBM 64 for some months now, I wonder if you could help with a number of problems I have encountered: 1) How can I get the computer variable to 2 decimal places? 2) The operator's manual says that you can define strings to 256 characters. When I try entering these, I get a "Syntax error" if I press the Return key after I have typed more than two lines of text. 3) Is there a simple way of using the 8 function keys in a program written in BASIC or can they only be defined in machine code?
Yours faithfully,
Lester Knight
West Sussex

We answer,
Printing numbers to two decimal places is a common problem but there is an easy solution. Firstly we need a function to round off a number. This means that 1.554 becomes 1.55 and 1.555 becomes 1.56. Here is the function (define it near the start of your program):

```
10 deffnro(z)=int(z★100+.5)
/100
```

Now here is a subroutine which will format numbers to two decimal places (adding a '-' sign at the start if it is negative). Note that the format string that this subroutine returns (a$) is always the same length (10 characters) so that a list of numbers will always be neat.

```
500 b=abs(fnro(a):a$=str$(b+.001):ifb=0thena$=".00 "
510 rem                    don't forget this space
520 a$=mid$("-   ",sgn(a)+2,1)+left$(right$("      "+a$,10),9)
530     rem      minus space space             6 spaces
540 return
```

You can only define strings to 255 characters. If you are typing a string into the computer you must remember that when the computer inputs from the screen, it only inputs two adjoining lines of 40 characters each. So if you were typing a long string into a line then when you had finished the computer would look at the last two lines and not find a line number and would therefore produce ?syntax error. The way to create long strings is by concatenation — adding strings together.

Each of the function keys has an ASCII value so if you wrote a program using the GET command and pressed one of these keys, the ASCII value of that key would be returned. The keys are not definable from BASIC.

Dear Sir,
I own a Commodore VIC 20 and have recently received a Commodore 1541 disc drive with which I seem to be having some problems. After I have run some of my programs I've found that when I give the drive a command it comes up with "Device not present error"; but the drive is connected and is switched on. To get the drive working again I have to Run Stop and Restore the computer and then give the command: OPEN 15,8,15,"I" (a warm start for the drive). It must be something in the program that somehow switches the drive off. Could you please tell me what it is and how to overcome it? Also can you expand the buffer so that you have more than 10 characters - if so, how? Any help would be greatly appreciated.
Yours faithfully
Stuart Young
Victoria, Australia

We answer,
To solve your problems with the drive, you must send the drive a 'uk' or 'ul' command ie open 1,8,15,"uk":closel. This must be done when you first switch it on and every time the drive is reset. The command tells the drive what speed to run it at — either for a VIC 20 or for a CBM 64.

The keyboard buffer may not be safely extended on the VIC 20 or the CBM 64.

Dear Sir,
I am writing a project program for my 'A' level computer studies exam. However, a difficulty has arisen which I would be grateful for some help with. My program runs on a Commodore PET 4032. I require a program allowing me to run and print graphics, eg bar charts, on the PET 4032 compatible printer. This screen dump is essential for my program. I urgently require a listing of a suitable screen dump on the PET 4032.
Yours faithfully,
David Horton
Sheffield

We answer,
Information on a Commodore screen is stored in screen code. This means that if you poke a screen location with zero an @ will appear — coincedentally this is the same as chr$(0). If you poke a screen location with 1 an 'a' will appear but chr$(1) is a reverse (or control) 'a' and this would send your printer in to enhanced mode. This subroutine can be converted to run on any of the Commodore machines printing to a Commodore dot matrix printer. The listing here is for a 4032:

```
10  nc=40:nr=25:rem number of columns : number
of rows
20 sc=32768 :rem memory
location of start of screen
30 open4,4
40 for i = 1 to nr-1 : a$="" :
for j = 1 to nc-1
50 a = peek( sc+i★nc+j) :b=0
60  if  a  and  128  then
a$=a$+chr$(18)  :  a = a and
127: b=1
70 if a < 32 then a = a or 64 :
goto 90
80 if a>63 then a = a or 128
90 a$ = a$chr$(a): if b then
a$ = a$chr$(146)
100 next : print£4,a$ : next :
close 4
```

As you can see, the program is quite straight forward and can be easily coded into machine code if required.

Dear Sir,
I have a Commodore 64 computer and also the EP-22 typewriter/printer from Brother, but I have tried without success to get the appropriate interfacing cable. I shall be very grateful if you could kindly supply me with any information on such cables.
Yours faithfully
J Lee
Manchester

We answer,
The interface and cable for the Brother EP22 printer is available from:

Scan Products
41, River Lane
Gaywood
Kings Lynn
Norfolk
PE30 4HD Tel: (094573) 581

It costs £19.50 and is suitable for the CBM 64 and for the VIC 20.

You could well drive yourself completely round the bend with this exciting game for the unexpanded VIC 20 from Andrew Laycock.

TRACK KING

ARE YOU SITTING COMFORTABLY? Is your seat belt firmly fastened and your nerves steady? Then you can begin! Use your skill as a driver to reach the end of the race track and still be alive! You have to choose the correct times to accelerate and decelerate or you could end up crashing into the barriers or into the back of another car.

Key to success

The game is played under the direction of the keyboard: U slows down the car, H moves the car to the left, J moves the car to the right and N increases your speed. By keeping the U key pressed, you will slow down more each time you move and so if you want to slow down by only the slightest fraction, press U once and then take your finger off the key. The same principle applies while speeding up using the N key.

Every time the level changes, the track becomes longer making it harder for you to reach the end!

The details

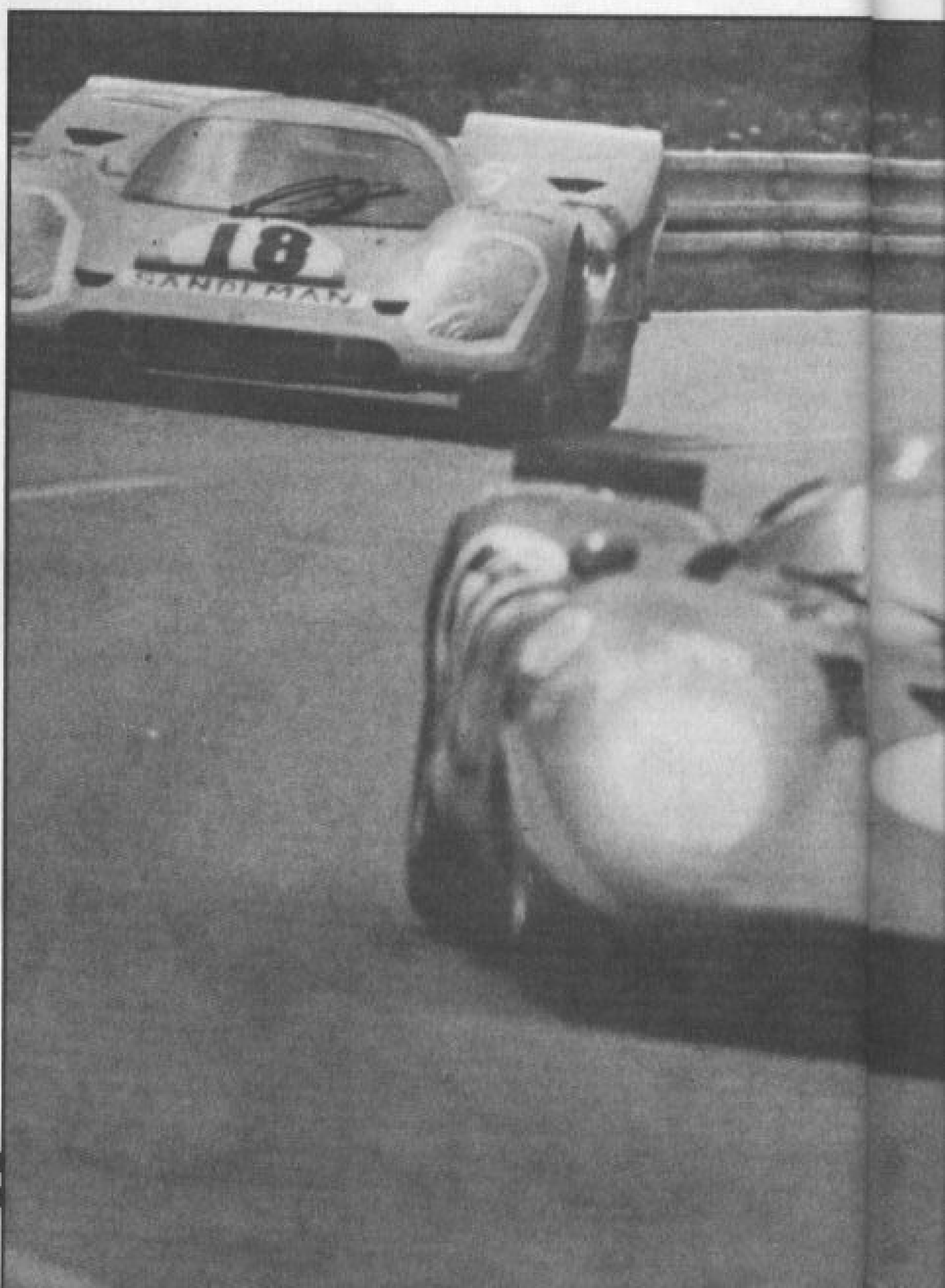
The game is made up of three programs. Program 1 consists mainly of PRINT statements although some POKEs are involved. Program 2 defines the graphics. Program 3 contains the nitty-gritty stuff and the variables and line functions are given:

Line explanation

LINE	FUNCTION
1-8	Initialise variables:first screen
9-12	Scroll screen in and out.
13	Set screen ready for game.
14	Print sides of track.
15-16	Variables.
20-39	Main section of program.
40-42	Crash routine.
49	Used with Line 41
50-61	Player reaches end: Prints what level he is at.
1000	Sound for every move.
2000	Used for crash graphics.
3000-3006	Sound for ambulance when crash occurs.

Variable explanation

VARIABLE	FUNCTION
SC	Score
H1	Highest score
LG	Level
TC	Finishing distance
TA	Distance travelled
TK	Your car
T	30720
SP	Speed
S1	36878: sounds, etc
P1	Check for key pressed
L and M	Cars that move towards you
KT	Checks your car.



Program 1

```
1 PRINT"";
2 P=0:P2=-1
3 F=1
4 ONFGOSUB10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27
5 FORT=PTOP2STEP1:PRINTTAB(T);A$;:FORY=0TO100:NEXTY:PRINTTAB(T);" ":NEXT
6 PRINTTAB(T);A$:PRINT"";:F=F+1
7 POKE36878,15:P3=P+237
8 POKE36876,P3:FORT=0TO10:NEXTT:POKE36878,0:IFK=0THEN4
9 GOTO30
10 A$="-":P=P+1:P2=P2+2:RETURN
11 A$="-":P=P+1:P2=P2+1:RETURN
12 A$="Y":P=P+1:P2=P2+1:RETURN
13 A$="O":P=P+1:P2=P2+1:RETURN
14 A$="U":P=P+1:P2=P2+1:RETURN
15 A$="R":P=P+1:P2=P2+1:RETURN
16 A$=" ":P=P+1:P2=P2+1:RETURN
17 A$="C":P=P+1:P2=P2+1:RETURN
18 A$="O":P=P+1:P2=P2+1:RETURN
19 A$="M":P=P+1:P2=P2+1:RETURN
20 A$="M":P=P+1:P2=P2+1:RETURN
21 A$="O":P=P+1:P2=P2+1:RETURN
22 A$="D":P=P+1:P2=P2+1:RETURN
23 A$="O":P=P+1:P2=P2+1:RETURN
24 A$="R":P=P+1:P2=P2+1:RETURN
25 A$="E":P=P+1:P2=P2+1:RETURN
26 A$="-":P=P+1:P2=P2+1:RETURN
27 A$="-":P=P+1:P2=P2+1:K=1:RETURN
30 PRINT"";
31 FORT=0TO17:PRINTTAB(T);"";:FORY=0TO100:NEXTY:POKE36878,15:POKE36876,T+237
32 FORH=1TO10:NEXTH:POKE36878,0:NEXT:PRINT"";
33 FORT=0TO17:PRINTTAB(T);"_";:FORY=0TO100:NEXTY:POKE36878,15:POKE36876,T+237
34 FORH=1TO10:NEXTH:POKE36878,0:NEXT:FORG=0TO10:NEXTG:GOSUB200:PRINT"
";
38 PRINT"";
39 FORT=2TO17:READD$:PRINTTAB(T);D$;:FORY=0TO100:NEXTY
40 POKE36878,15:POKE36876,255-T:FORH=0TO10:NEXTH:POKE36878,0:NEXT:GOSUB46
41 DATA"*",*,"* ",T,R,A,C,K," ",K,I,N,G," *",*,*
42 FORT=0TO21:READD$:POKE36878,15:POKE36876,T+234:FORH=1TO10:NEXTH:POKE36878,0
43 IFD$="O"THEN47
44 PRINTD$;:FORY=0TO100:NEXTY:NEXT
45 DATA"",Y,,,"",":    * ",A,N,D,R,E,W," L",A,Y,C,O,C,K," *",O
46 PRINT"";:RETURN
47 FORT=0TO10:NEXT
48 PRINT"";
49 FORT=0TO11:READD$:PRINTD$;:FORY=0TO200:NEXT:POKE36878,15:POKE36876,128+T:FORH
=1TO10
50 NEXTH:POKE36878,0:NEXT
51 DATA"P",R,E,S,S," A",N,Y," K",E,Y,":"
52 FORT=0TO10:NEXT
53 POKE198,0:WAIT198,1:POKE198,0
54 FORT=0TO11:READD$:PRINTD$;:POKE36878,15:POKE36876,244+T:FORH=1TO10:NEXT:FORY=
0TO30
55 POKE36878,0:NEXTY:NEXTT
56 DATA"     I",N,S,T,R,U,C,T,I,O,N,S
57 FORT=0TO21:PRINT"";TAB(T);"-";:FORY=0TO100:NEXTY:POKE36878,15:FORH=1TO10
58 POKE36876,255-T:NEXTH:POKE36878,0:NEXTT
59 FORT=0TO14:READD$:PRINT"";TAB(T);D$:FORY=0TO30:NEXTY:POKE36878,15:FORH=1
TO10
60 POKE36876,128+T:NEXTH:POKE36878,0:NEXTT
61 DATA"U",S,E," ",T,H,E,S,E," ",K,E,Y,S,":"
62 PRINT" = DECCELERATE"
63 FORT=790TO840:POKE36878,15:POKE36876,PEEK(T):NEXT
64 PRINT" = LEFT":FORT=790TO840:POKE36876,PEEK(T):NEXT
65 PRINT" = RIGHT":FORT=790TO840:POKE36876,PEEK(T):NEXT
66 PRINT" = ACCELERATE":FORT=790TO840:POKE36876,PEEK(T):NEXT
67 PRINT"      HIT";:GOSUB68:GOTO69
68 FORT=840TO790STEP-1:POKE36876,PEEK(T):NEXT:RETURN
69 PRINT" A";:GOSUB68
70 PRINT" KEY.":GOSUB68:POKE36876,0
71 POKE198,0:WAIT198,1:POKE198,0
72 PRINT"";:R=1
73 P=2:P2=P+3:P3=P
74 ONRGOSUB80,81,82,83,84,85,86,87,88,89
75 FORT=PTOP2:PRINTTAB(T);A$;:FORY=0TO150:NEXTY:FORH=1TO10:POKE36876,A:NEXTH
76 PRINTTAB(T);" ":POKE36876,0:NEXT
77 PRINTTAB(T);A$:PRINT"";
78 R=R+1:P=R+1:P2=P2+1:IFA$="G"THEN90
79 GOTO74
80 A$="T":A=128:RETURN
81 A$="R":A=138:RETURN
82 A$="A":A=148:RETURN
83 A$="C":A=158:RETURN
84 A$="K":A=168:RETURN
85 A$=" ":A=178:RETURN
86 A$="K":A=188:RETURN
87 A$="I":A=198:RETURN
88 A$="N":A=208:RETURN
89 A$="G":A=218:RETURN
90 T=0
91 PRINT"";:GOSUB92:GOTO93
92 FORT=6TO15:PRINTTAB(T);"*";:FORY=0TO50:NEXT:NEXT:RETURN
93 PRINT"";:GOSUB92:PRINT"";
94 PRINT"PLEASE";:FORT=790TO840:POKE36876,PEEK(T):NEXT:POKE36876,0
95 PRINT" LOAD";:FORT=790TO830:POKE36876,PEEK(T):NEXT:POKE36876,0
96 PRINT" PROGRAM";:FORT=790TO830:POKE36876,PEEK(T):NEXT:POKE36876,0
97 PRINT" 2":FORT=840TO790STEP-1:POKE36876,PEEK(T):NEXT:POKE36876,0
100 REM***************
101 REM END PROG (1)
102 REM***************
103 PRINT""
104 STOP
200 PRINT"  ___________    |   PRESENTS -   |   ___________
":RETURN
```



Program 2

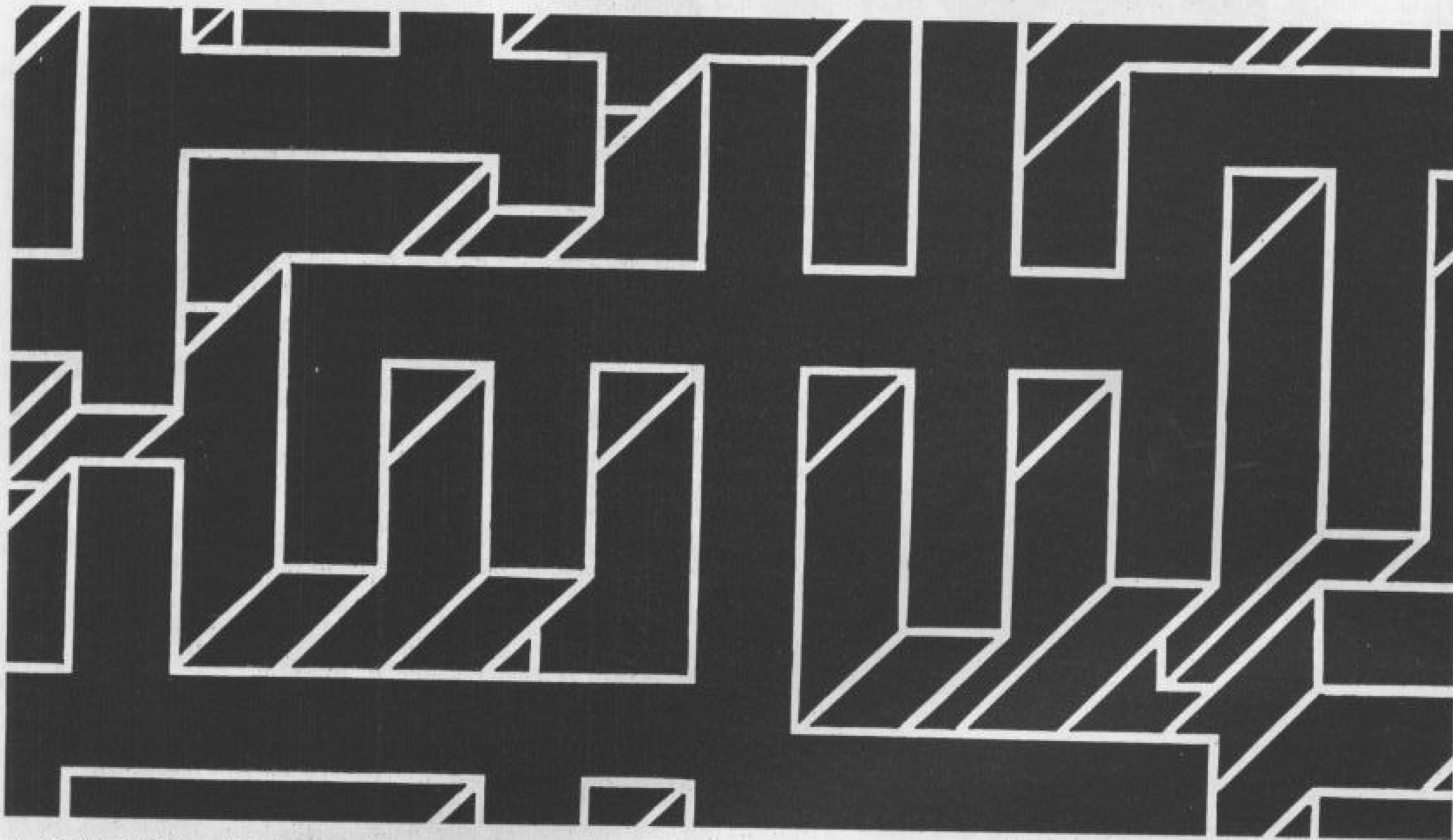
```
0 PRINT"PLEASE WAIT."
1 FORT=0TO511:POKET+7168,PEEK(T+32768):NEXT
2 FORI=0TO207:READD:POKEI+7168,D:NEXTI
3 DATA255,255,255,255,255,255,255,255,129,129,129,129,129,129,129,255
4 DATA84,4,161,9,0,74,2,145,84,4,161,9,0,74,2,145
5 DATA255,255,192,192,192,192,192,192,192,192,192,192,192,192,255,255
6 DATA255,255,195,195,195,195,195,195,255,255,192,176,216,204,198,195
7 DATA255,255,195,195,195,195,195,255,255,195,195,195,195,195,195,195
8 DATA255,255,192,192,192,192,192,255,255,3,3,3,3,3,255,255
9 DATA195,195,195,195,195,195,195,255,255,195,195,195,195,195,195,195
10 DATA24,24,24,24,24,24,24,24,24,24,24,24,0,0,24,24
11 DATA165,165,255,255,165,165,60,60,60,60,90,90,126,126,66,66
12 DATA255,255,192,192,192,192,192,255,255,192,192,192,192,192,255,255
13 DATA255,255,195,195,195,195,195,195,195,195,195,195,195,195,195,195
14 DATA252,252,195,195,195,195,195,195,195,195,195,195,195,195,254,254
15 DATA165,165,255,255,165,165,60,60,60,60,90,90,126,126,66,66
16 PRINT"PLEASE";:GOSUB17:GOTO18
17 FORT=790TO840:POKE36878,15:POKE36876,PEEK(T):NEXT:POKE36876,0:RETURN
18 PRINT" LOAD";:GOSUB17
19 PRINT" PROGRAM";:GOSUB17
20 PRINT" 3":GOSUB17
30 PRINT"";:STOP
```

Program 3

```
1 SC=0:HI=0:LG=1
2 POKE36869,240:POKE36879,27:POKE36865,24:POKE36867,44:TC=2000:TA=0
3 PRINT"      TRACK KING"
4 PRINT"-----------------------"
5 PRINT"YOUR DISTANCE=";SC
6 PRINT"LONGEST EVER =";HI
7 PRINT"      HIT A KEY."
8 POKE198,0:WAIT198,1:POKE198,0:PRINT"";
9 FORZX=0TO22STEP1:POKE36864,12+ZX:POKE36865,38+ZX:POKE36866,150-ZX:POKE36867,17
4-ZX*2
10 POKE36879,ZX:NEXTZX
11 FORZX=22TO0STEP-1:POKE36864,12+ZX:POKE36865,38+ZX:POKE36866,150-ZX:POKE36867,
174-ZX*2
12 POKE36879,ZX:NEXTZX:POKE36879,24
13 POKE36865,20:POKE36869,255:POKE36867,33:POKE36879,24:PRINT""
14 FORXZ=0TO22:PRINT"":NEXT
15 TK=7756:T=30720
16 SP=400:S1=36878
20 POKETK,32:POKETK+T,1
30 P1=PEEK(197):IFTA=TCTHEN50
31 IFP1=43THENTK=TK-1
32 IFP1=20THENTK=TK+1
33 IFP1=51THENSP=SP+20
34 IFP1=28THENSP=SP-20
35 L=INT(RND(1)*6)+4:M=INT(RND(1)*7)+10
36 PRINT"":PRINTTAB(L);"":PRINTTAB(M);"H":TA=TA+1
37 KT=PEEK(TK):SC=TA:IFSC>HITHENHI=SC
38 IFKT=0ORKT=8THENGOSUB2000:GOTO40
39 POKETK,12:POKETK+T,2:FORUI=0TOSP:NEXTUI:GOSUB1000:GOTO20
40 PRINT""
41 FORT=250TO150STEP-1:POKE36874,T:GOSUB49:POKE36876,T:GOSUB49:NEXT:POKE36876,0
42 POKE36874,0:GOSUB3000:GOTO2
49 FORTY=0TO10:NEXT:RETURN
50 PRINT"":LG=LG+1
51 POKES1,15
52 FORT=150TO250STEP1:POKE36874,T:GOSUB49:POKE36876,T:NEXT:POKE36876,0:POKE36878
,0
60 POKE36869,240:POKE36867,44:PRINT"LEVEL ";LG;" GAME":TC=TC+2000:T
A=0
61 FORTP=0TO4000:NEXT:POKE36874,0:POKE36876,0:GOTO9
1000 POKES1,15:POKES1-2,255:FORLI=0TO10:NEXT:POKE36876,0:RETURN
2000 POKETK,1:POKETK+T,0:RETURN
3000 POKE36878,15
3001 FORT=15TO0STEP-1
3002 POKE36876,200:FORU=0TO500:NEXT:PRINT""
3003 POKE36876,0
3004 POKE36875,232:FORU=0TO500:NEXT:PRINT""
3005 POKE36875,0:POKE36878,T:NEXT
3006 RETURN
```

You have to cope with mazes (rather than snakes) with the ladders in this great game from Andrew Laycock.

LADDER MAZE



A COMPASS MAY BE OF some help in this game, although I very much doubt it, and I'm not sure that a ball of string would add much to your chances of success! You must guide your man (could his name be Vic by any chance?) through the maze and up the ladders.

The U key moves your man up the ladders, the H key moves him to the left and the J key moves him to the right. Ratings are given on the time it takes you to reach the top so use your skills to get the best rating you can and see if you can rise to the challenge!

To explain. . .

Program 1 is an instructions program made up almost totally of PRINT statements.

Program 2 can be broken down as follows:

Program 2 explanation

LINE	USE
0	Starts at 1000
1-9	PRINTs maze lines and first ladder. Sets some variables.
10-22	Main part of game movements, etc.
50	"You made it". Player reaches top.
100-110	Reprint ladders. Choose new position for ladder.
1000-1006	Read data for graphics. Set screen
2000-2004	When player reaches top, send him back down.
3000-3010	Screen between games.
4000	Sets Y$
6000-6005	Ratings.

The variables used are:

A and X	Used to set first screen: X = ladders.
OP	Used to make man fail.
Z	Sets first ladder.
M	Man
TI$	Time
U	Checks for key being pressed
HJ	Check if new ladder is needed
N1	Used in PEEK for man.
M$	Substitutes for TI$

Listing 1

```
1 PRINT"■     LADDER MAZE."
2 PRINT"■  BY. ANDREW LAYCOCK"
3 PRINT"-----------------------"
4 PRINT"  THIS GAME TESTS YOUR■SKILL AS YOU -GUIDE- A■MAN THROUGH A MAZE AND■UP
";
5 PRINT"LADDERS."
6 PRINT"■ WHEN YOU HAVE CLIMBED■TO THE TOP YOU WILL BE■GIVEN A RATING ON YOUR■SK
ILL."
7 PRINT"■       HIT A KEY."
8 POKE198,0:WAIT198,1:POKE198,0
9 PRINT"■      LADDER MAZE"
10 PRINT"■-----------------------"
11 PRINT"■USE THESE KEYS:"
12 PRINT"■ ■■■■ = UP                    "
13 PRINT"  ■■■■ = LEFT                 "
14 PRINT"  ■■■■ = RIGHT                "
15 PRINT"       HIT A KEY."
16 POKE198,0:WAIT198,1:POKE198,0
17 PRINT"■     LADDER MAZE.     ■-----------------------■"
18 PRINT"PLEASE";:POKE36878,15:FORT=790TO840:POKE36876,PEEK(T):NEXT:POKE36876,0
19 PRINT" LOAD";:FORT=790TO840:POKE36876,PEEK(T):NEXT:POKE36876,0
20 PRINT" PROGRAM";:FORT=790TO840:POKE36876,PEEK(T):NEXT:POKE36876,0
21 PRINT" 2":FORT=840TO790STEP-1:POKE36876,PEEK(T):NEXT:POKE36878,0:PRINT"■■■■■"
```

Listing 2

```
0 TI$="":RUN1000
1 PRINT"■";
2 A=7702:X=8142:POKE36878,15
3 B=30720
4 P=A+21:OP=4
5 FORC=ATOP:POKEC,0:POKEC+B,2:POKE36876,234+C-A:NEXT:POKE36876,0
6 A=A+44:IFA=8186THEN8
7 GOTO4
8 Z=INT(RND(1)*21)+8142:POKEZ,2:POKEZ+22,2:POKEZ+B,0:POKEZ+22+B,0
9 M=8164:TI$="000000"
10 POKEM,32:POKEM+B,1
11 U=PEEK(197)
12 IFU=51THENM=M-22:HJ=HJ+1:POKE36876,255:FORJL=0TO10:NEXT:POKE36876,0
13 IFPEEK(M)=N1THENM=M+22:HJ=0:GOTO11
14 IFHJ=2THENGOSUB100:OP=OP+2
15 IFU=43THENM=M-1:POKE36876,232:FORJL=0TO10:NEXT:POKE36876,0
16 IFPEEK(M)=N1THENM=M+1
17 IFU=20THENM=M+1:POKE36876,200:FORJL=0TO10:NEXT:POKE36876,0
18 IFPEEK(M)=N1THENM=M-1
19 POKEM,1:POKEM+B,4
20 IFM<7702THEN50
21 FORF=0TO30:NEXT
22 PRINT"■               ":PRINT"■TIME =";TI$:GOTO10
50 PRINT"■■YOU MADE IT      ":FORPO=0TO1000:NEXT:GOSUB4000:M$=TI$:GOTO2000
100 IFGH=0THEN110
101 POKEX+L,2:POKEX+L+22,2:POKEX+L+B,0:POKEX+L+22+B,0
102 L=INT(RND(1)*21):X=X-44
103 POKEX+L,2:POKEX+L+22,2:POKEX+L+B,0:POKEX+L+22+B,0:HJ=0
104 POKE36876,234+L:FORKL=0TO10:NEXT:POKE36876,0:RETURN
110 POKEZ,2:POKEZ+22,2:POKEZ+B,0:POKEZ+22+B,0:GH=1:          GOSUB102:RETURN
1000 FORT=0TO511:POKET+7168,PEEK(T+32768):NEXT
1001 FORT=0TO23:READD:POKET+7168,D:N1=0:NEXT
1002 POKE36869,255
1003 RUN1
1004 DATA255,129,153,153,153,153,129,255
1005 DATA24,60,24,255,24,24,36,36
1006 DATA129,129,255,129,129,255,129,129
2000 FORFG=0TOOP
2001 POKEM,1:POKEM+B,U
2002 FORYI=0TO10:POKE36877,252:NEXTYI:POKE36877,0:U=U+1
2003 IFU=8THENU=0
2004 POKEM,32:M=M+22:NEXT:GOSUB4000
3000 PRINT"■";:POKE36869,240
3001 PRINT"      LADDER MAZE       "
3002 PRINT"-----------------------■";:PRINTY$
3003 PRINT"YOUR TIME WAS ";M$:PRINT"■■":GOSUB6000
3004 PRINT"■DO YOU WANT ANOTHER    ■GAME (Y/N)"
3005 GETR$:IFR$=""THEN3005
3006 IFR$="Y"THENPOKE36869,255:RUN1
3007 IFR$="N"THEN3009
3008 GOTO3005
3009 POKE36869,240:FORT=0TO511:POKET+7168,PEEK(T+32768):NEXT
3010 NEW
4000 Y$="■":RETURN
6000 IFM$<"000010"THENPRINT"RATING = SUPER FAST.":RETURN
6001 IFM$<"000015"THENPRINT"RATING = VERY FAST.":RETURN
6002 IFM$<"000020"THENPRINT"RATING = FAST.":RETURN
6003 IFM$<"000030"THENPRINT"RATING = FAIRLY SLOW.":RETURN
6004 IFM$<"000060"THENPRINT"RATING = VERY SLOW.":RETURN
6005 IFM$>"000060"THENPRINT"ANY SLOWER AND YOU'LL     ■   STOP.":RETURN
```

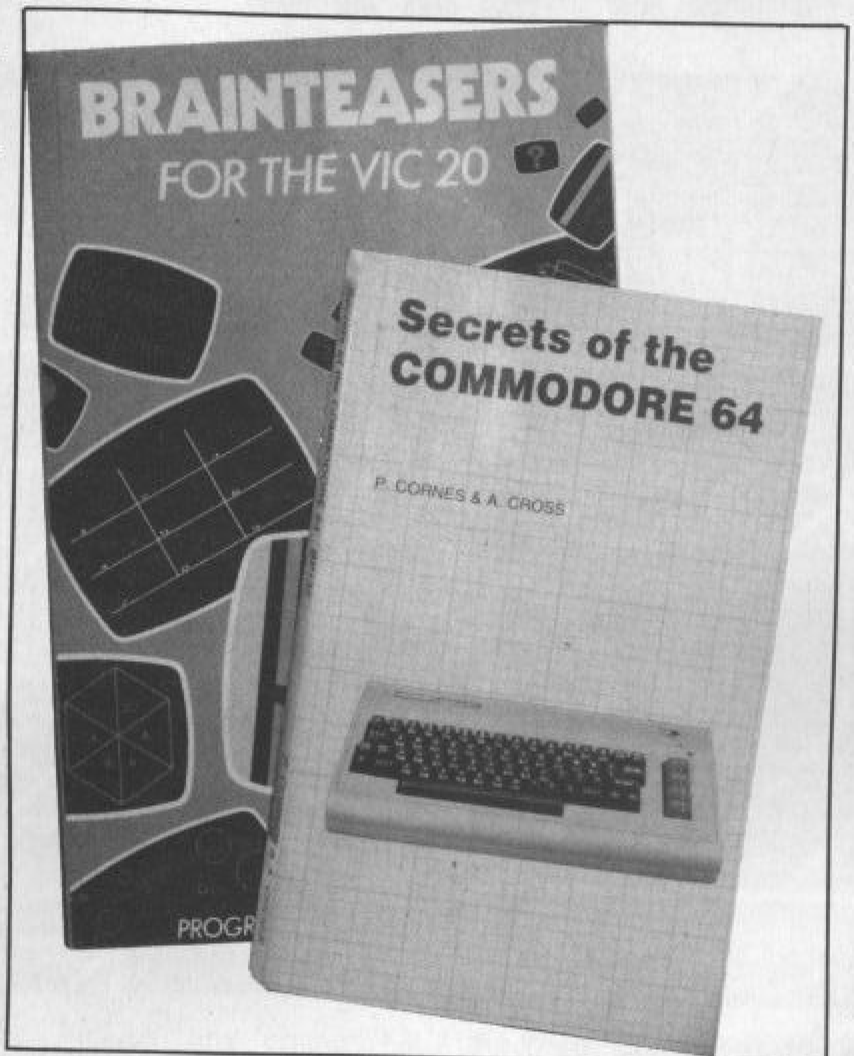
We have a look at some of the books sitting on our shelves.

REFERENCE LIBRARY

Book Title:
Secrets of the Commodore 64
Author:
P. Cornes and A. Cross
Publisher:
Bernard Babani Ltd.
Price:
£1.95

THIS POCKET-SIZED manual professes to be yet another beginners' guide to the Commodore 64. Although the book will win no prizes for originality, for those of you who haven't the time or the inclination to plough your way through some of the more weighty volumes on the market, it might prove invaluable as a brief yet informative introduction to the Commodore 64.

The book commences by informing us, for the umpteenth time in the history of computer literature, how the Commodore 64's memory is organised and of the proportion of that memory which is available to the Commodore 64 from BASIC. Some knowledge of the fundamental concepts of the Commodore 64 is obviously presumed since the authors ignore the simpler facets of the BASIC programming language by launching straight into such relative complexities as the all-important random numbers and means of generating them, the PEEK and POKE statements, the GET statement and input routines. The reader is also enlightened on character graphics, sprites and high resolution graphics. Chapters on sound, in the realms of both sound effects and music creation, and machine code programs and the use of machine code statements to extend BASIC, conclude this book.



On the whole, the book is user-friendly with chapters divided into short sections of succinct, jargon-free explanations elucidated by several diagrams and code-listings in easy-to-read bold type.

Book Title:
Brainteasers for the VIC 20
Author:
Genevieve Ludinski
Publisher:
Phoenix Publishing Associates
Price:
£5.95

GENEVIEVE LUDINSKI, AN experienced programmer and technical author with her own software company specialising in educational programs, has produced this book of twenty-two brainteasers for those VIC 20 users whose idea of fun is to tear their hair out and bite their nails while testing their mental agility. Many of the programs even contain an IQ rating at the end of the program, assessing the speed and dexterity with which you solve the problem in hand.

The brainteasers utilize the machine's music and graphics capabilities. Ms. Ludinski explains her puzzles briefly and intelligibly and the code for the brainteasers is listed in a clear and readable form.

A fairly wide selection of puzzles are offered ranging from run-of-the mill magazine games (Word Search) and maze games (Wire Maze and A-Maze-ing (not that title again!)) to musical puzzles (Saints to Sinners — a musical guessing game) and the obligatory space adventure, Close Encounters. The Francis Drake Adventure Game, the program rated by Ms. Ludinski as her star performer, lets the reader step back in history and into the shoes of Sir Francis Drake as he sailed off in pursuit of the North West Passage. You also have the chance to be on either side of the law. For the villains amongst you, the book offers Safe Cracker, a simple guessing game complete with police sirens if an incorrect guess leaves the cops hot on your trail and Western Adventure which allows you to live the life of bandits on the run from the sheriff after robbing a bank, in true western style. Or put on your detective hat for Detective or an Agatha Christie style Who Dunnit.

So whether you see yourself as cop or robber, musician or explorer, this book provides the reader and VIC 20 user with puzzles to tax the brain while stepping into the shoes of his alter ego; the less adventurous amongst you can stick to Ms. Ludinski's number and word puzzles.

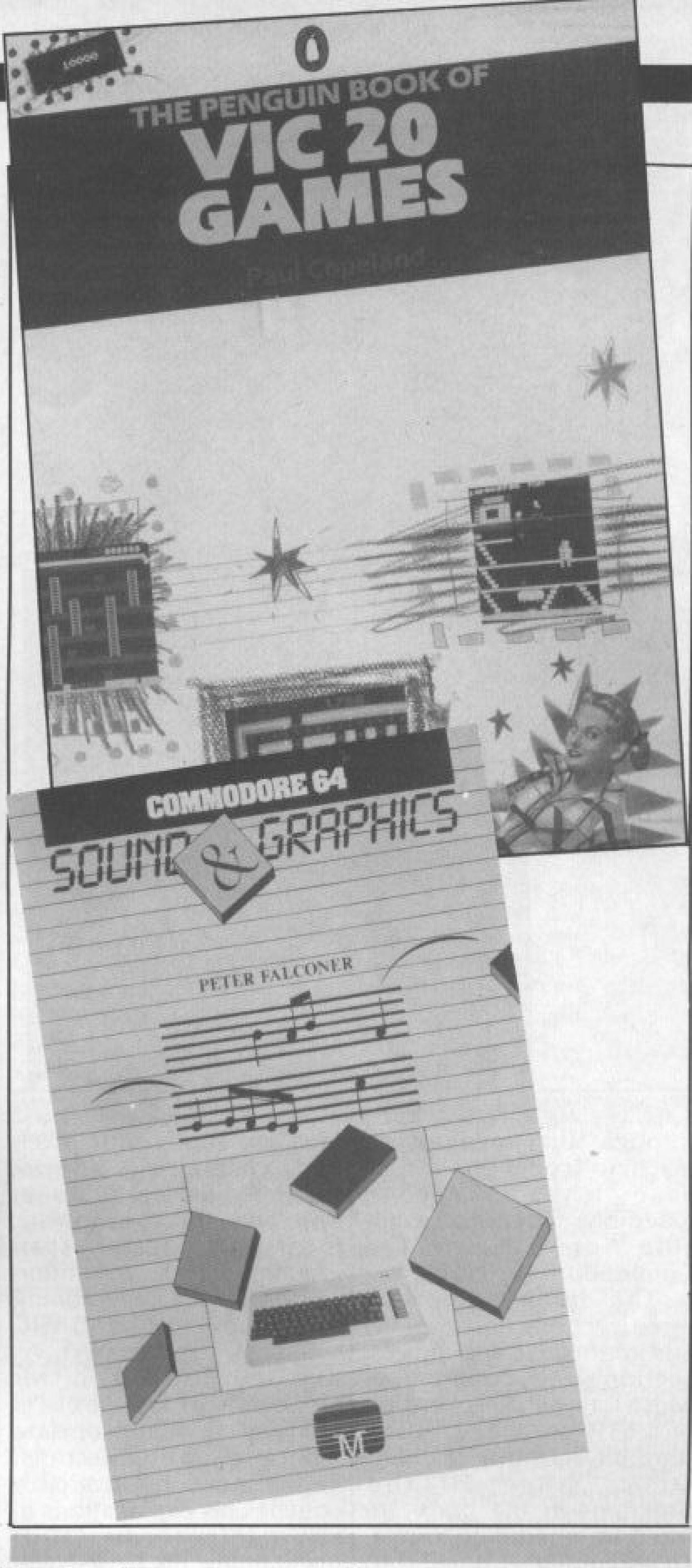
Book Title:
Commodore 64 Sound & Graphics
Author:
Peter Falconer
Publisher:
Melbourne House Publishers

THE COMMODORE 64'S sophisticated sound and graphic capabilities, covered briefly in the

Commodore 64 manual, are expounded in this book by Peter Falconer. The features available are explained and the user is involved in the design and coding of a real application. The game used for the purpose of illustration throughout the book is written in both BASIC and machine code, thus enabling the user to exploit the full potential of the Commodore's sound and graphic features. But don't fret if you're yet to discover the delights of machine code since Mr. Falconer presents his machine code routines in three ways: as assembly language code, as BASIC programs illustrating the same idea (where possible) and as DATA statements for inclusion in BASIC programs. The author has squeezed a lot of useful information into a small book and has made this information easy to follow by presenting it in small blocks of text interspersed with examples of code. The book is divided into two sections, the first on low resolution graphics (ie, character graphics) and the second, smaller section, on bit mapped graphics. As per usual, the user's ability to follow the code is hampered by the illegible Commodore graphic symbols.

The book starts with a simple game program written in BASIC. This game is enhanced throughout the book by the addition of graphics, sound and more professional presentation. These features are covered chapter by chapter. The mysteries of sprites, music and interrupts are explained. Sound on the Commodore (voices, waveforms, volume, for example) is explored and the more complex facilities of the Commodore 64 such as scrolling, sprites and high resolution graphics are investigated.

Having pieced the game together chapter by chapter, the user should be left with the know-how to use the Commodore 64's graphic and sound features when developing his own games and applications.



Book Title:
Business Systems on the Commodore 64
Author:
Susan Curran and Margaret Norman
Publisher:
Granada
Price:
£6.95

THIS BOOK, WRITTEN BY Susan Curran, a full-time writer specialising in computers and their applications, and Margaret Norman, a freelance writer and computer programmer, provides a practical and informative introduction to the Commodore 64 as a small business computer. It shows how the Commodore 64, low priced yet well established and highly reliable (a crucial factor when considering hardware for business applications), can be employed to run business application packages.

The book starts with a general explanation of the Commodore 64, assessing why it is a good machie for the small business and painting a broad picture of the business software available. It does not profess to be a maintenance guide or programming guide nor is it merely a lengthy list of business applications available for use on the Commodore 64. Each chapter commences with some background information into the type of application under discussion. The text is clearly illustrated with computer printouts and screen displays. The reader is told how transition from manual accounting, stock-taking, or whatever the business application in question may be, to computerisation may complement his business. The types of software available for a particular application are then discussed.

The applications included are computerised accounts, stock control and other money-oriented programs, spreadsheets, databases and filing systems, word processing programs and a final chapter on applications excluded from this list.

To conclude, "Business Systems on the Commodore 64", provides a clear and concise introduction to the different types of business applications which may be run on the Commodore 64 and explains how they may be integrated into the small business.

Book Title:
The Penguin Book of VIC 20 Games
Author:
Paul Copeland
Publisher:
Penguin
Price:
£2.95

PAUL COPELAND'S BOOK of VIC 20 games, at £2.95, is good value for money. The book contains an assortment of adventures, arcade-type games and board games. The author's musical background is reflected in the sophisticated musical routines in some of the games, Life, Al-Khwarizm and Music Sequencer, for example. As well as ten games, the book includes chapters on utility programs and creating your own

games. The author claims to have fully exploited the colour, sound and graphics potential of the VIC 20.

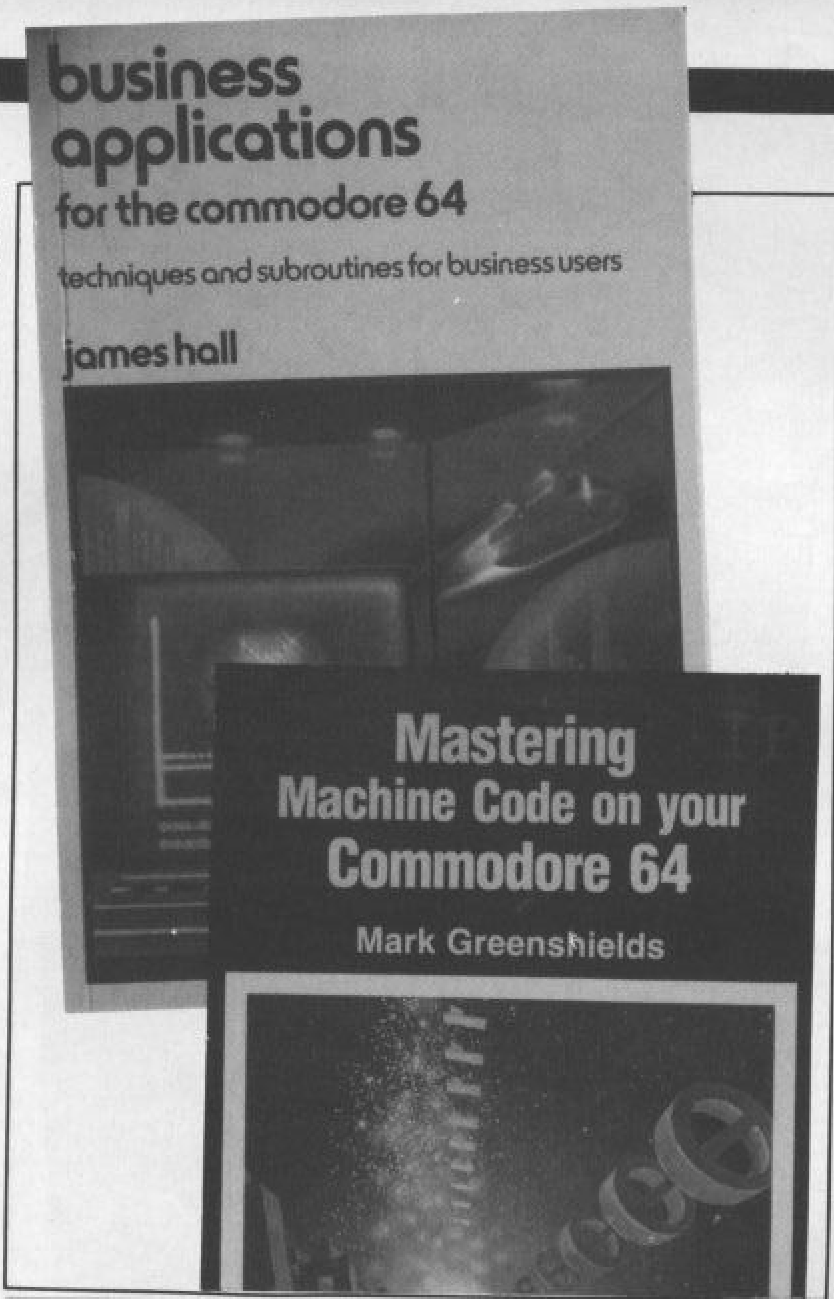
The first two games, You Draw and Noughts and Crosses, may be adapted for use on disc based systems. All the games are introduced by a brief synopsis of what should happen if you enter the program code correctly and by descriptions of the program structure. The game listings conclude the chapters.

The games vary from the commonplace — Noughts and Crosses — to the adventurous such as Red Alert where, as commander-in-chief of a space ship you are sent on a secret mission into outer space to place various space modules in quadrants of space.

The entrepreneurs amongst you may wish to test your skills at Oil Rig whereby, as director of an oil mining company, you have to find as many oil fields as possible (up to a maximum of ten). If your forte lies in gambling rather than business, step back to the Las Vegas of 1965 where your target is to break the bank before it breaks you.

Less philistine readers may wish to take their creative talents to the small screen with You Draw or Music Sequencer.

A set of handy utility routines and a group of scenarios to help you create your own games can be found at the end of the book.



Book Title:
Mastering Machine Code on your Commodore 64
Author:
Mark Greenshields
Publisher
Interface Publications
Price:
£7.95

THIS GLOSSY LITTLE volume, written by the author of "Mastering the Commodore 64", allows the Commodore user, already conversant in BASIC, to create more efficient, faster and professional programs through the medium of machine code — in this case, the 6502/6510 Assembly language, available for use on the Commodore 64.

The book comprises three sections and numerous informative appendices. Section one commences with a listing and explanation of a 6510 assembler/dissassembler/monitor, SUPERMON. Since all the programs in the book are listed in mnemonic format, an assembler is needed to enter them. This section continues with explanations of every 6510 Assembly language command and every programming mode of the 6510 chip. The tutorial is interspersed with copious and lucid examples of Assembly language programs.

The second section shows the user, having now achieved some proficency in Assembly language programming, how to capitalise on his newly acquired skills by putting them to practical use. The use of Assembly language in scrolling (both with pixels and characters), sprites, music (scales and tunes on one and three channels), interrupts, raster scan graphics, high resolution graphics and, finally, adding commands to the BASIC language, are covered. As with the first section, Mr Greenshields' text is clarified by appropriate examples and listings. Furthermore, he invariably couches his explanations in lay-man's terms, thus maintaining his back-cover promise to produce a book for the Assembly language beginner.

The third and final section covers the ROM routines inside the Commodore 64 and instructs the user on how best to apply them to their programs.

The book ends with a generous spread of appendices (12 in all) many of which appear to have been reprinted from the Commodore 64 manual (ASCII values for characters and control codes.

Book Title:
Business Applications for the Commodore 64
Author:
James Hall
Publisher:
Sunshine Books
Price:
£5.95

FOR THOSE COMMOdore 64 users who wish to use their computer as a business machine without turning to the off-the-shelf business applications market, this book provides a useful introduction to designing and writing bespoke business software. In his introduction, Mr. Hall exposes the amount of time which may be saved in changing from a manual to a computerised system, as deduced from his own experience. Moreover, he states that there is no need to invest in expensive peripherals to run business programs on the Commodore 64 (apart from the use of a printer for some applications such as word processing) although he does, later on, stress the advantage of a disc drive in providing greater memory capacity and faster file access for business applications. The inexperienced programmer need not fret since these programs are accompanied by step-by-step documentation and illustrations where necessary.

The book introduces the reader to business program design with a synopsis of computers and file handling. The applications covered by the book are constructed from a library of subroutines which can then be amended, added to and tailored for the user's individual needs. A useful final chapter allows the user to enter a routine to replace dot matrix graphic symbols with letters and also includes a utility program.

So, if your Commodore 64 is employed as a business machine and you are brave enough to exploit the advantages of creating your own business applications, this attractive book is a wise investment.

VIC GAMES PROGRAMMING

This great series introduces you to the delights of programming games for the VIC in BASIC. Bryn Phillips takes us through the steps.

THIS IS THE FIRST OF A FIVE part series of BASIC games programming for the VIC 20. The series is primarily intended for newcomers to games programming, but there might well be a few useful tips for seasoned programmers.

Many people are put off writing action games in BASIC because they think that it is too slow, and have read somewhere or other that any good game must be written in Machine Code. That's true up to a point. A badly written full scren version of Space Invaders written in BASIC could be really painful. Semi-comatose Aliens jerking down the screen being potted off one by one with your laser in between cups of coffee would not be exactly inspiring. On the other hand a well written Lunar Lander or Shooting Gallery game could rival many of the commercial products. In fact it would be better, because it would be YOUR game with your own amazing graphics and sound effects, and that is what it's all about.

In this series I will attempt to show you some of the techniques which you can use to write effective games programs in BASIC, and if you follow it through and use your imagination you should even be able to do something about that Space Invaders game!

Screen display

This month the emphasis is on the screen display, and this can be one of the most enjoyable aspects of writing a program. It's a good place to start because speed doesn't matter — the action doesn't start until you've set up your play area. The key point is to get it looking good, and that takes planning. One way of doing this is to draw it out on paper first. Graph paper is OK, but don't forget your final screen display will be rectangular. The only problem with using paper is that it is time consuming and discourages experimentation. Once you've got something which looks reasonable you tend to stick with it, and you're immediately off to a bad start. It's much better to use the screen to experiment, and the utility program "SCREEN DESIGNER" included in this article will help you to do this. It's worth noting at this point that the VIC 20 has an extremely well thought out graphics set built into it. Unfortunately there is a great temptation to forget this and always use your own user defined graphics (we'll be covering this in Part 4). But don't ignore it — even if you use your own graphic characters you can always incorporate some of the standard graphics into your own set. When you use this program pay particular attention to the colours. Often an otherwise good program is ruined by murky graphics. This is because some of the character and screen colour combinations just don't mix very well, to the extent that some are virtually undecipherable. It

Listing 1

```
10 PRINT"♥"
20 PRINT" ********************"
30 PRINT" *                  *"
40 PRINT" *                  *"
50 PRINT" *                  *"
60 PRINT" * **************** *"
70 PRINT" *                  *"
80 PRINT" *                  *"
90 PRINT" *                  *"
100 PRINT" * **************** *"
110 PRINT" *                  *"
120 PRINT" *                  *"
130 PRINT" *                  *"
140 PRINT" * **************** *"
150 PRINT" *                  *"
160 PRINT" *                  *"
170 PRINT" *                  *"
180 PRINT" * **************** *"
190 PRINT" *                  *"
200 PRINT" *                  *"
210 PRINT" *                  *"
220 PRINT" ********************"
230 PRINT"■■MEMORY FREE"FRE(0)
240 GOTO240
```

```
********************
*                  *
*                  *
*                  *
* **************** *
*                  *
*                  *
*                  *
* **************** *
*                  *
*                  *
*                  *
* **************** *
*                  *
*                  *
*                  *
* **************** *
*                  *
*                  *
*                  *
********************
```

Figure 1.

varies from computer to computer and your TV can also make a difference. Play about with the graphics, make some shapes, design a screen — EXPERIMENT!

The program

So now you've got a screen design and want to put it in your program. There are two options in BASIC on the VIC 20, PRINTing or POKEing. Let's look at a simple screen display, such as the one in fig. (1).

You can either PRINT it or POKE it onto the screen. PRINTing is the tempting option as you simply write a number of PRINT statements containing the design. This is shown in Listing 1.

In contrast POKEing a design onto the screen takes a bit more thought. You have to define your screen co-ordinates and use the appropriate POKE formulae. The simplest method is to regard the screen in terms of X,Y co-ordinates and define the bottom left hand corner as 0,0. To POKE a character onto the screen use:

POKE P1+X–22★Y,CH

and to POKE a colour use:

POKE P2+X–22★Y,CL

Where P1=8164
P2=38884

Listing 2 gives an example of how the screen design can be POKEd onto the screen.

At first sight PRINTing is better as it's much faster. Wait a minute though, speed doesn't matter here. What about memory? If you look at the free memory displayed you will see that Listing 1 only left 2929 Bytes free, whereas Listing 2 left 3262 Bytes free, a saving of 233 Bytes, and that could be crucial later on. If you examine Listing 2 you will see that it could be tightened up even further by using multiple statement lines, GOSUBs etc, and this is shown in Listing 3 where a further 88 Bytes have been saved, a total saving of 315 over Listing 1.

If you're writing games programs on an unexpanded VIC 20, tight programming is essential. Admittedly the listings don't look so good, the program can be difficult to follow and debug, but you leave yourself enough memory over for the game and with any luck will have enough left over for all those frills that make it look professional — High score record, menu option, keybord/joystick capability and of course the sound effects. As you will see in a future article tight programming also means spedier action, and it really is worth taking the time to plan ahead, and to practice squeezing everything down as far as you can.

Random thoughts

Finally let's have a look at the RND function. Often it can be used to very good effect in screen design and in a multi-level game can be used to add variety as the game progresses. You can use it to change character

Listing 2

```
10 PRINT"♥"
20 P1=8164:P2=38884
30 FORX=1TO20
40 FORY=1TO21STEP20
50 POKEP1+X-22*Y,42
60 POKEP2+X-22*Y,6
70 NEXTY
80 NEXTX
90 FORX=3TO18
100 FORY=5TO18STEP4
110 POKEP1+X-22*Y,42
120 POKEP2+X-22*Y,6
130 NEXTY
140 NEXTX
150 FORY=1TO20
160 FORX=1TO20STEP19
170 POKEP1+X-22*Y,42
180 POKEP2+X-22*Y,6
190 NEXTX
200 NEXTY
210 PRINT"■ MEMORY FREE"FRE(0)
220 GOTO220
```

Listing 3

```
10 PRINT"♥"
20 P1=8164:P2=38884
30 FORX=1TO20:FORY=1TO21STEP4:GOSUB100:NEXTY,X
40 FORY=1TO21:FORX=1TO20STEP19:GOSUB100:NEXTX,Y
50 FORY=2TO20:FORX=2TO19STEP17:POKEP1+X-22*Y,32:NEXTX,Y
60 PRINT"■ MEMORY FREE"FRE(0)
70 GOTO70
100 POKEP1+X-22*Y,42:POKEP2+X-22*Y,6:RETURN
```

Listing 4

```
10 PRINT"♥"
20 P1=8164:P2=38884
30 CH=160:CL=0
40 FORX=1TO20:FORY=1TO21STEP20:GOSUB100:NEXTY,X
50 FORY=1TO21:FORX=1TO20STEP19:GOSUB100:NEXTX,Y
60 CH=102:CL=2
80 GOSUB90:GOSUB100:GOSUB90:GOSUB110:GOTO80
90 X=INT(RND(1)*18+2):Y=INT(RND(1)*19+2):GOSUB100:RETURN
100 POKEP1+X-22*Y,CH:POKEP2+X-22*Y,CL:RETURN
110 POKEP1+X-22*Y,32:RETURN
```

Screen Designer Program

```
10 REM SCREEN DESIGNER          FOR ANY VIC 20
20 REM BRYN PHILLIPS            1983
24 REM
25 REM REPEAT ALL KEYS
26 REM
30 POKE650,128
40 CH=32:CL=0
44 REM
45 REM ADJUSTS FOR MEMORY SIZE
46 REM
50 P1=484+4*(PEEK(36866)AND128)+64*(PEEK(36869)AND112)
60 P2=484+37888+4*(PEEK(36866)AND128)
70 PRINT"■■■■■SCREEN DESIGNER "
80 PRINT"■■■KEYS HAVE NORMAL       VALUES APART FROM:"
90 PRINT"■■CHANGE COLOURS(1-8)"
100 PRINT"■F1 - CHARACTER"
110 PRINT"■F3 - BORDER"
120 PRINT"■F5 - SCREEN"
130 PRINT"■■CLR - TRANSPARANT"
140 PRINT"■■F8 - RE-RUN"
150 PRINT"■■USE CURSOR CONTROLS    TO MAKE DESIGN"
160 PRINT"■■PRESS C TO CONTINUE"
170 GET A$:IFA$<>"C"THEN 170
180 PRINT"■"
185 B=8:S=1:GOSUB490
190 POKEP2+X-22*Y,0
200 GETA$
204 REM
205 REM CH=19 IS CLR KEY (TRANSPARANT)
206 REM
210 PR=CH:IFCH=19THENPR=PEEK(P1+X-22*Y):CL=PEEK(P2+X-22*Y)
220 POKEP1+X-22*Y,131:POKEP2+X-22*Y,0:POKEP1+X-22*Y,PR:POKEP2+X-22*Y,CL
230 IFA$=""THEN200
240 A=ASC(A$)
250 IFA=17ORA=29ORA=145ORA=157THENGOSUB360:GOTO350
254 REM
255 REM A=18 IS RVS ON/A=146 IS RVS OFF
256 REM
260 IF A=18 THEN RV=128:GOTO350
270 IF A=146THENRV=0
280 IFA=133THENGOSUB430:GOTO350
290 IFA=134THENGOSUB460:GOTO350
300 IFA=135THENGOSUB510:GOTO350
310 IFA=140THENRUN
320 A=A-64:IFA<0THENA=A+64
330 IFA>128THENA=A-64
340 CH=A+RV
350 GOTO200
354 REM
355 REM MOVE CURSOR
356 REM
360 IFA$="■"THENX=X+1:IFX>21THENX=21
370 IFA$="■"THENX=X-1:IFX<0THENX=0
380 IFA$="■"THENY=Y+1:IFY>22THENY=22
390 IFA$="■"THENY=Y-1:IFY<0THENY=0
400 PR=CH:IFCH=19THENPR=PEEK(P1+X-22*Y):CL=PEEK(P2+X-22*Y)
410 POKEP1+X-22*Y,PR:POKEP2+X-22*Y,CL
420 RETURN
424 REM
425 REM CHANGE CHARACTER COLOUR
426 REM
430 GETC$:IFC$=""THEN430
440 CL=ASC(C$)-49:IFCL<0ORCL>7THEN430
450 RETURN
454 REM
455 REM CHANGE BORDER COLOUR
456 REM
460 GETB$:IFB$=""THEN460
470 B=ASC(B$)-41
480 IFB<8ORB>15THEN460
490 SC=16*S+B
500 POKE36879,SC:RETURN
504 REM
505 REM CHANGE SCREEN COLOUR
506 REM
510 GETS$:IFS$=""THEN510
520 S=ASC(S$)-49
530 IF S<0ORS>7THEN510
540 SC=16*S+B
550 POKE36879,SC:RETURN
```

positions, characters and colours, and is particularly useful in maze designs. A simple example is shown in Listing 4.

Type in the program and RUN it. The program simply draws a black boundary around the screen, and randomly POKEs red squares onto the screen. Now comes the important part. Sit back and watch the screen. At this point it is possible that you will just see squares appearing and disappearing. If this is the case you need to try harder. Relax, and look again. It could be the start of a meteor storm scenario, or a devilish hyper-galactic maze, or moles appearing and disappearing in your garden! Remember, we'll be able to make those squares look like anything, and you can change the ratio of squares to blanks, mix colours, add extra characters etc. It could be the basis of a good game. We'll have another look at it next month.

Next time. . .

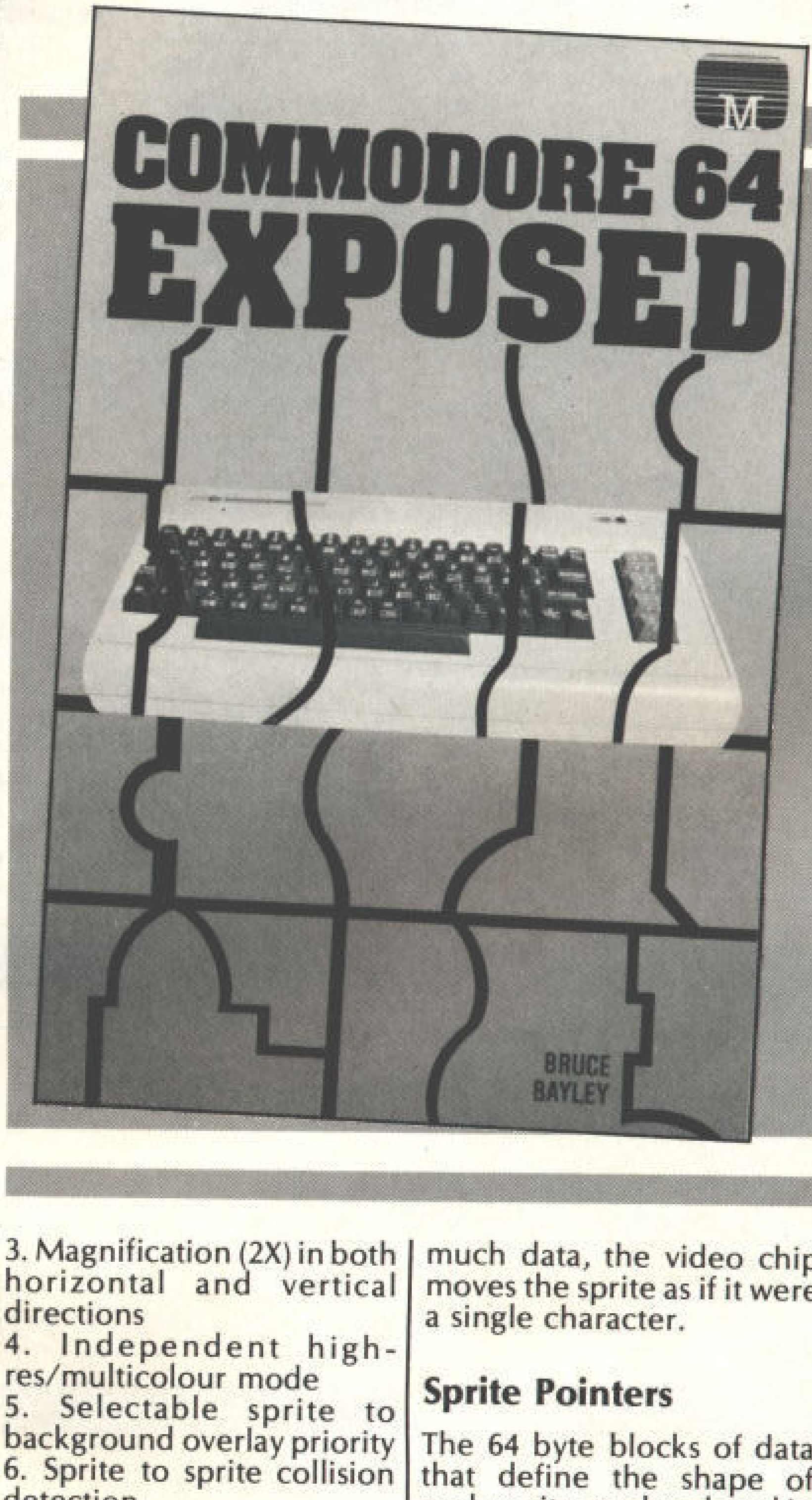
In the next part of this series we'll be looking at getting the characters moving, using programmed commands, and keyboard or joystick interaction. In other words we'll get it all moving.

Screen designer

This program will allow you to experiment with screen designs. On running the program you will be presented with a blank screen containing a flashing cursor which you can move around the screen using the cursor controls. If you press a key then the appropriate characer will be entered on the screen elsewhere the cursor goes. You can use the full VIC character set, including reversed characters. If you want a reversed character press CTRL/RVS, and the next character entered from the keyboard will be reversed. You can change the colour of the character by pressing F1, followed by the appropriate colour key, and similarly change the border and screen colours by using F3 and F5 respectively.

Just as a change from the usual type of book review, we thought we'd show you exactly the sort of thing you'll be getting for your money.

COMMODORE 64 EXPOSED



THIS ARTICLE IS AN extract from the Graphics chapter of Melbourne House's Commodore 64 Exposed by Bruce Bayley, which we are reprinting with their kind permission. This should give you, not just a lot of useful information on sprites, but an idea of the high quality of this book. Commodore 64 Exposed costs £6.95 in paperback, and contains 195 pages.

Melbourne House are concentrating heavily on the Commodore 64 this autumn. They have just published Commodore 64 Sound and Graphics, and at the end of September Commodore 64 Machine Language for the Absolute Beginner will be available. Both books will also be priced at £6.95.

Melbourne House books can be found in all good computer bookshops, or they can be contacted directly at Castle Yard House, Castle Yard, Richmond, TW10 6TF.

Sprites

A sprite is a form of user defined character that is controlled by a powerful video chip called the 6566. Up to 8 sprites can be displayed at a time automatically. More sprites can be displayed using Raster Interrupt techniques. Sprites have the following advantages over user defined characters:

1. Pixel by pixel movement in any direction.
2. The 24 by 21 pixel sprite shape can be moved as though it were a single character
3. Magnification (2X) in both horizontal and vertical directions
4. Independent high-res/multicolour mode
5. Selectable sprite to background overlay priority
6. Sprite to sprite collision detection
7. Sprite to background collision detection.

A sprite is larger than a character, therefore more data is needed to define the shape of a sprite. A sprite is 24 pixels (3 bytes) wide and 21 pixels high which gives us a total of 3 x 21 = 63 bytes of data to define the shape of a single sprite. Even though a single sprite is made up of so much data, the video chip moves the sprite as if it were a single character.

Sprite Pointers

The 64 byte blocks of data that define the shape of each sprite can be placed in any 64 byte multiple of unused memory. In order to tell the video chip where in memory each sprite-shape block is located, eight sprite pointers are provided.

The shape of a sprite may be changed by adjusting the sprite pointer allocated to that sprite to point to a different block of sprite-shape data. Using this method a single sprite may be animated by quickly changing the sprite's pointer to switch through a series of shapes provided for that sprite's animation (e.g. an explosion). Switching the pointer rather than switching between sprites leaves the other sprites free for other uses.

The sprite pointers are the last 8 bytes of unused screen memory (2040 — 2047). If you move screen memory, the pointers will move with it (but not their contents). You must remember when setting up your sprite pointers that the pointer must point to the first byte within the sprite and that the value in the sprite pointer is the actual memory location of the sprite over 64. Therefore, the following formula applies:

Location = Sprite pointer ★ 64

Also if you are not using video bank # 0 (default bank) then you must also add bank number ★ 16384 to the location. If you haven't switched video banks then don't worry.

Two important points to remember when choosing where to put your sprite data in memory are 1, its location **must** be a multiple of 64, and 2, check the memory map to make sure that you are only using spare memory.

Turning Sprites On

For a sprite to be displayed to the screen, it must be turned on. The memory location where the video chip gets its information on which sprites should be

turned on and which should be turned off is location 53269. The 8 bits within byte 53269 are labled from right to left 0—7. Therefore, if we label our sprites from 0 — 7 then we easily determine which sprites should be on and which should be off by the value contained in byte 53269. The way that the on/off status of each sprite is determined is as follows:

A 1 in the bit corresponding to the sprite determines that the sprite should be displayed (turned on) and a 0 determines that the sprite should not be displayed (turned off).

e.g. 7 6 5 4 3 2 1 0
1 1 0 1 0 1 1 1= 215

therefore the statement POKE 53269, 215 would supply the video chip with the following information:
Sprites 7,6,4,2,1 and 0 are to be turned on.
Sprites 5 and 3 are to be turned off.
To turn on a single sprite without effecting the others, use the following statement:
POKE 53269, PEEK (53269) OR (2↑SN)
where SN is the sprite number (0 — 7)
To turn off a single sprite without effecting the others, use the following statement:
POKE 53269, PEEK (53269) AND (255 — 2↑SN)

Sprite Colour

High resolution (single colour) sprites can be any one of the 16 colours. The colour of each sprite 0 — 7 should be POKEd into their respective colour registers, memory location 53287 — 53294 (see video register map). Each pixel turned on within the sprite will be displayed in the colour determined by the sprite's colour register. Each pixel turned off will be displayed in the colour behind the sprite (i.e. it is transparent).

Multicolour Sprites

In multicolour mode, it is possible to have four different colours in each sprite. Though, as with multicoloured characters, multicoloured sprites have only half the resolution of single coloured sprites (ie. pixels must be displayed in pairs). The following table gives the colours determined by each bit-pair combination.

Bit pair	Resultant Colour
00	Transparent(screen colour)
01	Sprite multicolour register #0 (location 53285)
10	Sprite-colour register
11	Sprite multicolour register #1 (location 53286)

The register that holds information on which sprites are multicoloured and which sprites are not is mapped to location 53276.

To set a sprite to multicolour, use the following statement:
POKE 53276, PEEK (53276) OR (2↑SN)
where SN is the sprite number (0 — 7).
To switch a sprite out of multicolour mode, use the following statement:
POKE 53276, PEEK (53276) AND (255 — 2↑SN)

Expanding Sprites

Sprites can be expanded vertically, horizontally or both. A sprite is expanded by putting 2 pixels in place of 1 and 2 blanks in place of 1 in the direction of expansion thus giving a 2X expansion. To expand a sprite horizontally, the corresponding bit in location 53277 must be set to 1. To reduce the sprite, the bit must be set to 0. Vertical expansion is don in the same way using location 53271. The POKE statements to control expansion and reduction of sprites are as follows:

2
Horizontal expansion
POKE 53277, PEEK (53277) OR (2↑SN)
Horizontal reduction
POKE 53277, PEEK (53277) AND (255 — 2↑SN)
Vertical expansion
POKE 53271, PEEK (53271) OR (2↑SN)
Vertical reduction
POKE 53271, PEEK (53271) AND (255 — 2↑SN)
where SN is the sprite number from 0 — 7.

Sprite Movement

Sprites are moved around the display by changing the values in each sprite's horizontal and vertical position registers. These registers are mapped to memory location 53248 to 53263 and a most-significant-bit (MSB) register at location 53264. The MSB register is used to rectify the problem of horizontal screen width. The MSB register works as follows. In order to gain pixel movement, the horizontal position register needs to be able to hold values from 0 to 299 (screen width). A single register can only hold values from 0 to 255 therefore we need at least one more bit to handle values up to 299. An extra bit (9th bit) would allow us control over positions 0 to 511. This is the purpose of the MSB register. The bits in the MSB register correspond to the sprite number (ie bit 0 for sprite 0, bit 1 for sprite 1, etc.) A register map of all sprite positioning registers is given overleaf. Note that horizontal positions 24 and 344 are the left and right boudaries of the screen. Sprites continue to move outside this range but cannot be seen.

It's about time we had a look at one of these sprites.

Study the first program and its comments. Type it in and run it.
Run the program; you should see a square sprite float across the screen.
•To expand the sprite in the horizontal and vertical directions before moving, add the following line:
105 POKE VIDEO + 29,1 : POKE VIDEO + 23,1
and run the program again.

The second program allows you to use the cursor keys to draw a sprite by editing DATA statements. Type RUN 1, then use the cursor keys to move around the DATA statements. Use the shift Q character to signify a pixel-ON and a full-stop to signify a pixel-OFF. When you have finished drawing your sprite, move the cursor to the top of the screen, then keep hitting the RETURN key until you have entered all of the DATA statements. Now type RUN, and the program will generate the sprites and the DATA statements needed to generate that sprite. To store these DATA statements, use the same method as you used on the last set of DATA statements.

Sprite Display Priorities

Sprite priority determines if the sprite should appear in front or behind another background. If the backgroud is another sprite, then the priority is fixed by the sprite's sprite number. Sprite 0 has the highest priority, sprite 1 has the next priority, and so on, up to sprite 7. For example, if sprite 0 and sprite 7 are positioned so that they cross each other, sprite 0 will be in front of sprite 7, though you would be able to see sprite 7 through sprite 0 (unless of course sprite 0 was a completely filled square). Sprite to background priority is more flexible in the way that each sprite can be set with priority above or below the background. The sprite to background priorities are controlled by the sprite priority register (memory location 53275). A 1 in the bit number corresponding to the sprite number will set that sprite with a lower priority than the background. A 0 in this bit position will give the sprite a higher priority than the background. By moving sprites back and forth over other objects, at the same time changing the sprite-background priorities, it is possible to make it look as if the sprites are moving in front and behind the object thus creating a three dimensional effect.

The third program overlays 8 sprites to demonstrate sprite priority.

Listing 2

```
0 GOTO10
1 PRINTCHR$(147):POKE53269,0:LIST29-50
10 PRINTCHR$(147):FORI=0TO63:POKE832+I,0:NEXT
15 POKE53280,6:POKE53281,6
20 GOTO 60
29 REM...012345678901234567890123
30 DATA "●..●●●●●●●●●●●●●●●●●●●●●"
31 DATA "●●..●●●●●●●●●●●●●●●●●●●●"
32 DATA "●.●..●●●●●●●●●●●●●●●●●●●"
33 DATA "●..●..●●●●●●●●●●●●●●●●●●"
34 DATA "●...●..●●●●●●●●●●●●●●●●●"
35 DATA "●....●..●●●●●●●●●●●●●●●●"
36 DATA "●.....●..●●●●●●●●●●●●●●●"
37 DATA "●......●..●●●●●●●●●●●●●●"
38 DATA "●.......●..●●●●●●●●●●●●●"
39 DATA "●........●..●●●●●●●●●●●●"
40 DATA "●.........●..●●●●●●●●●●●"
41 DATA "●..........●..●●●●●●●●●●"
42 DATA "●...........●..●●●●●●●●●"
43 DATA "●............●..●●●●●●●●"
44 DATA "●.............●..●●●●●●●"
45 DATA "●..............●..●●●●●●"
46 DATA "●...............●..●●●●●"
47 DATA "●................●..●●●●"
48 DATA "●.................●..●●●"
49 DATA "●..................●..●●"
50 DATA "●●●●●●●●●●●●●●●●●●●●●..●"
60 V=53248:POKEV+16,1:POKEV+1,50 :POKEV+21,
   1:POKEV+39,3:POKE2040,13
70 POKEV+23,1:POKEV+29,1
80 FORI=0TO20:PRINT1000+I;"DATA";:READA$:FORK=
   0TO2:T=0:FORJ=0TO7:B=0
90 IF MID$(A$,J+K*8+1,1)="●"THENB=1
100 T=T+B*2↑(7-J):NEXT:PRINT T;",";:POKE 832+
   I*3+K,T:NEXT:PRINT"█ ":NEXT
110 END
3000 PRINTCHR$(19):END
9000 SAVE"@0:SPRITE",8
9010 VERIFY"SPRITE",8
```

Listing 1

```
1 REM *** SQUARE ***
5 REM   *CLEAR THE SCREEN TO BLUE WITH A BLACK
BORDER
10 PRINTCHR$(147):POKE53280,0:POKE53281,6
15 REM  *SET SPRITE-POINTER #0 TO POINT TO
LOCATION 13*64=832
20 POKE 2040,13
25 REM  *CREATE A SQUARE SPRITE IN MEMORY
LOCATIONS 832 TO 832+63
30 FOR MEM=832 TO 834:POKE MEM,255:NEXT
40 FOR MEM=835 TO 889 STEP 3
50 POKE MEM,128:POKEMEM+1,0:POKEMEM+2,1
60 NEXT MEM
70 FORMEM=892TO894:POKEMEM,255:NEXT
75 REM  *SET BEGINNING OF VIDEO CHIP
80 VIDEO=53248
85 REM  *TURN ON SPRITE #0
90 POKE VIDEO+21,1
95 REM  *CHOOSE THE COLOUR WHITE FOR SPRITE #0
100 POKE VIDEO+39,1
109 REM *MOVE SPRITE ACROSS SCREEN
110 Y=100 : POKE VIDEO+1,Y:FOR X=0 TO 347
115 REM *CALCULATE X-POSITION AND MSB
120 MSB=INT(X/256) : XP=X-256*MSB
130 POKE VIDEO+0,XP: POKE VIDEO+16,MSB
140 NEXT X
```

Listing 3

```
5 REM * DEMONSTRATING EIGHT SPRITES *
10 POKE53280,6:POKE53281,0
20 PRINT CHR$(147);"CREATING SPRITES"
30 POKE 52,62 :POKE 56,62 :P=248
40 FOR MEM=2040TO2047:POKEMEM,P:P=P+1
50 NEXT MEM
51 :
60 BYTE=272
70 FOR SN=0 TO 7 : PRINT SN
80 LOC=PEEK(2040+SN)*64
90 BYTE=BYTE/2:B=BYTE:ROW=0
100 FORMEM=LOCTOLOC+63STEP3
110 IFSN<=3THEN140
120 ROW=ROW+1:T=(ROW-SN-3)/4:B=0
130 IF INT(T)=T THEN B=255
140 FOR COL=0TO2:POKEMEM+COL,B:NEXT COL
150 NEXT MEM,SN
155 :
160 VIDEO=53248:A=150:B=0
170 POKE VIDEO+21,255:POKE VIDEO+28,255
175 POKE VIDEO+37,1:POKE VIDEO+38,1
180 C=0:FOR R=39 TO 46:C=1
190 POKE VIDEO+R,C:PRINTCHR$(147):NEXTR
200 I=A:A=B:B=I:D=SGN(B-A)
210 FOR SN=7 TO 0 STEP-1
215 Z=11*(SN>3)*(A=0)
220 PRINT CHR$(147);"SPRITE";SN
230 SX=VIDEO+SN*2:SY=SX+1
240 Y=A-D+Z:FORX=A+ZTOB+ZSTEPD:Y=Y+D
250 POKE SX,X:POKE SY,Y
260 NEXT X,SN
270 FOR PAUSE=1TO2000:NEXT
290 GOTO 200
```

Sprite Collisions

Sprite collisions are detected by the computer and collision information is stored in location 53278 for sprite to sprite collisions and location 53279 for sprite to another background collision. The bit set to 1 in each of these registers corresponds to the sprite involved in the collision. The bit stays set until the register is read (PEEKed). So if the collision information is to be used more than once per collision, it would be a good idea to store the value into a variable. Also, programs that use the sprite collision registers should include in their initialization a PEEK of each of these registers to clear them of previous collision data.

Note: A bit pair 01 in a multicoloured mode will not be detected in a sprite to background collision, even though it can be seen on the screen. So, for example, if you wish to have objects that should not cause a collision (e.g. a cloud) then they should be coloured by using a bit pair 01 (multicolour register # 01)

Location	Use of Register
53248	Sprite 0 X position
53249	Sprite 0. Y position
53240	Sprite 1 X position
53251	Sprite 1 Y position
53252	Sprite 2 X position
53253	Sprite 2 Y position
53254	Sprite 3 X position
53255	Sprite 3 Y position
53256	Sprite 4 X position
53257	Sprite 4 Y position
53258	Sprite 5 X position
53259	Sprite 5 Y position
53260	Sprite 6 X position
53261	Sprite 6 Y position
53262	Sprite 7 X position
53263	Sprite 7 Y positon
53264	Sprite (0—7)MSB register

VIC-II Register Map

Reg.#		6	5	4	3	2	1	0	
0	Sprite # 0				x–position				
1	Sprite # 0				y–position				
2	Sprite # 1				x–position				
3	Sprite # 1				y–position				
4	Sprite # 2				x–position				
5	Sprite # 2				y–position				
6	Sprite # 3				x–position				
7	Sprite # 3				y–position				
8	Sprite # 4				x–position				
9	Sprite # 4				y–position				
10	Sprite # 5				x–position				
11	Sprite # 5				y–position				
12	Sprite # 6				x–position				
13	Sprite # 6				y–position				
14	Sprite # 7				x–position				
15	Sprite # 7				y–position				
16	Sprite # 7	Sprite # 6	Sprite # 5	Sprite # 4	Sprite # 3	Sprite # 2	Sprite # 1	Sprite # 0	MSB of x–position
17	Raster MSB	Extended colour	Bit map	Display enable	Screen height	—	Vertical scroll	—	Mode/y–scroll
18	Raster register								
19	Light pen–x								
20	Light pen–y								
21	Sprite # 7	Sprite # 6	Sprite # 5	Sprite # 4	Sprite # 3	Sprite # 2	Sprite # 1	Sprite # 0	Sprite enable
22	—	—	Reset Multi	Multi colour	Screen width	—	Horizontal scroll	—	Multi colour/x scoll
23	Sprite # 7	Sprite # 6	Sprite # 5	Sprite # 4	Sprite # 3	Sprite # 2	Sprite # 1	Sprite # 0	Sprite y–expand
24	Screen location				Character base			—	
25	IRQ	—	—	—	Light pen	Sprite-sprite collision	Sprite-b'gnd collision	Raster	Interrupt register
26	IRQ	—	—	—	Light pen	Sprite-sprite collision	Sprite-b'gnd collision	Raster	Interrupt enable
27	Sprite # 7	Sprite # 6	Sprite # 5	Sprite # 4	Sprite # 3	Sprite # 2	Sprite # 1	Sprite # 0	Sprite–background priority
28	Sprite # 7	Sprite # 6	Sprite # 5	Sprite # 4	Sprite # 3	Sprite # 2	Sprite # 1	Sprite # 0	Sprite multicolour select
29	Sprite # 7	Sprite # 6	Sprite # 5	Sprite # 4	Sprite # 3	Sprite # 2	Sprite # 1	Sprite # 0	Sprite x–expand
30	Sprite # 7	Sprite # 6	Sprite # 5	Sprite # 4	Sprite # 3	Sprite # 2	Sprite # 1	Sprite # 0	Sprite to sprite collision
31	Sprite # 7	Sprite # 6	Sprite # 5	Sprite # 4	Sprite # 3	Sprite # 2	Sprite # 1	Sprite # 0	Sprite to background collision
32	Screen border Colour								
33	Background Colour # 0								
34	Background Colour # 1								
35	Background Colour # 2								
36	Background Colour # 3								
37	Sprite multicolour # 0								
38	Sprite multicolour # 1								
39	Sprite # 0 Colour								
40	Sprite # 1 Colour								
41	Sprite # 2 Colour								
42	Sprite # 3 Colour								
43	Sprite # 4 Colour								
44	Sprite # 5 Colour								
45	Sprite # 6 Colour								
46	Sprite # 7 Colour								

Your COMMODORE

YOUR BEST INDEPENDENT COMMODORE MAGAZINE

SO YOU OWN A COMMODORE?

SO YOU'VE WRITTEN SOME PROGRAMS?

SO WHY HAVEN'T YOU SUBMITTED THEM TO US?

Your Commodore is always on the lookout for new material for publication and we know that there are thousands of intelligent, literate, innovative and creative Commodore owners out there, so why don't we get together?

If you have written an exhilirating game or an invaluable utility on your Commodore micro, share your talents with us and our readers by submitting your efforts and the form to the address below. All articles should be documented and type-written and should be accompanied by a printout of the program as well as a copy of the program on cassette or disc. All material should be original; if it is not chosen for publication, it will be returned to you.

You may not have written any software yourself, but you have very firm opinions about the world of Commodore and all their attendant industries and products. Then put your opinions on paper and post them to us, again at the address below — you never know, you might even get paid for airing your views! All submissions should be sent to:

The Editor
Your Commodore
Argus Specialist Publications Limited
No 1 Golden Square
London WIR 3AB

★PLEASE COMPLETE IN BLOCK CAPITALS

Your Name

Program Name

Computer/memory size it runs on

Amount of memory program occupies

Other computers/memory size which your program runs on without conversion or use

Does your game need or use joysticks? Yes No

Have you sent our game to another magazine Yes No

Is it original/or a variation on a theme?

Your Address

Telephone Number

Times to contact you



64 Tape Computing adds a new dimension to your micro!

Run this Argus 64 tape and you'll soon see why it's Britain's top selling tape magazine. Each issue gives you a variety of exciting and challenging games to play, reviews of other newly released software plus valuable utilities enabling you to write your own programmes and games.

Stretch your imagination and skills with 64 Tape Computing – available every other month from WH Smith, Menzies and other leading stores.

(You'll see them advertised on TV from September!)

Get your copy today!

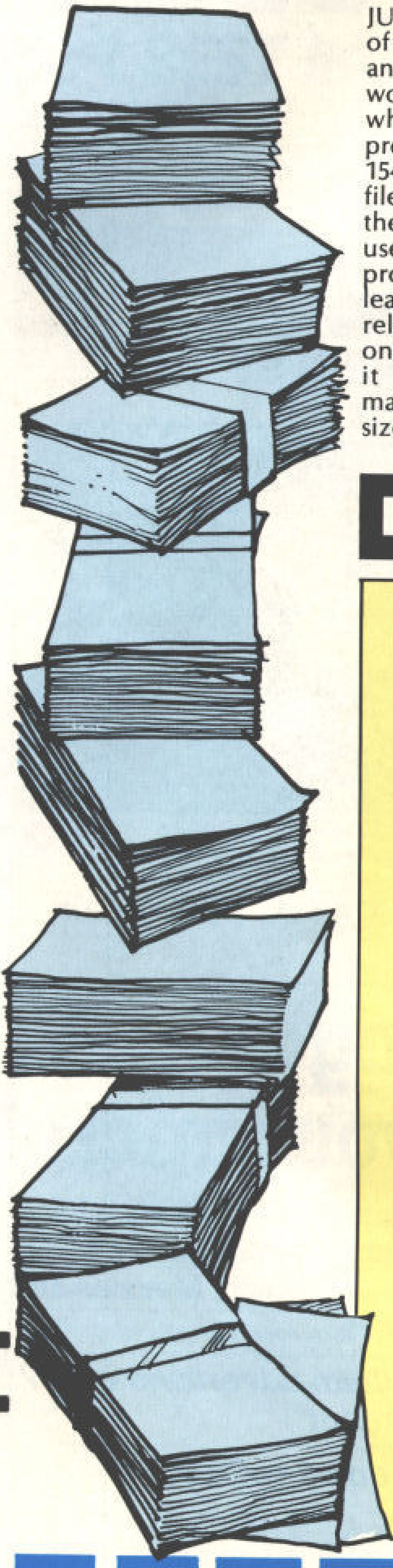
Argus Press Software Group

Argus Tape Magazines produced by
ARGUS PRESS SOFTWARE
1 Golden Square, London W1R 3AB
Telephone: 01 437 0626

Dave Crisp helps out in a retail outlet and he helps here with relative files, which seem to cause a lot of bother.

RELATIVE FILES

RELATIVE FILES



JUDGING BY THE AMOUNT of telephone calls, enquiries and questions I receive it would seem that one thing which causes a lot of problems to owners of the 1541 disc drive is relative files. Except for a few things the manual is more or less useless. Using these two programs you should at least be able to get BASIC relative files working and once you have got the idea it is easy to start to manipulate files and file sizes etc.

Relatively speaking. . .

Think of a relative file as a large array. Instead of the array being in RAM though it is stored in numbered order on the disc. As with an array it is easy to find a particular record if you know the number. If the number is not known it is possible to search the disc record by record.

The two programs are very basic. They have simple screen displays, there is very little error checking, and virtually no string handling. Once you have got the idea of what is going on then you can go back to the manual, which should then make more sense and tidy up the programs.

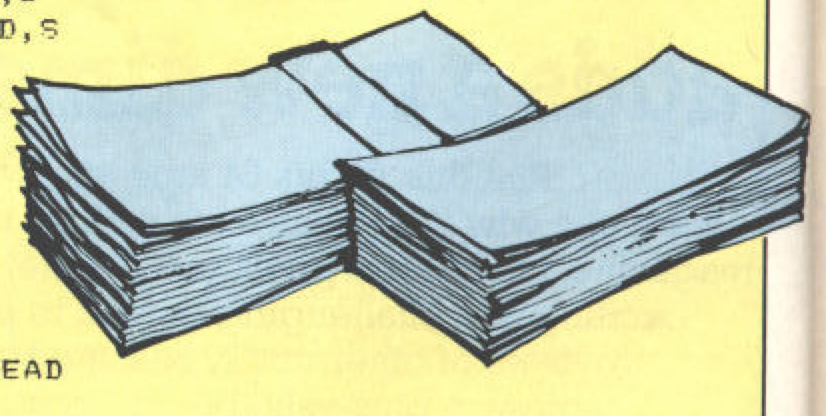
When creating the file for the first time, I recommend that you use a blank disc. This will ensure that all record numbers are available. Initially create a file of 254 characters/record and have about 100 records. It is worth noting that it is

Program 1 Listing

```
READY.

1 REM***** PROGRAM 1******
2 PRINT"▮"
5 INPUT"▮▮▮▮▮▮▮▮▮HOW MANY CHARACTERS/RECORD";RC
6 INPUT"▮▮▮▮HOW MANY RECORDS MAX.500";RM
10 OPEN15,8,15:OPEN2,8,2,"0:REL FILE,L,"+CHR$(RC)
20 FORRN=1TORM
30 X$="*"
40 RH=INT(RN/256):RL=INT(RNAND255)
45 PRINT"RH = ";RH,"RL = ";RL;
50 PRINT£15,"P"CHR$(2)CHR$(RL)CHR$(RH)CHR$(1)
60 PRINT£2,X$:S=ST
65 INPUT£15,A,B$,C,D
70 PRINTX$,A,B$,C,D,S
250 NEXT
260 CLOSE15:CLOSE2

READY.
```



```
SEARCHING FOR BIGREAD
LOADING

READY.
```

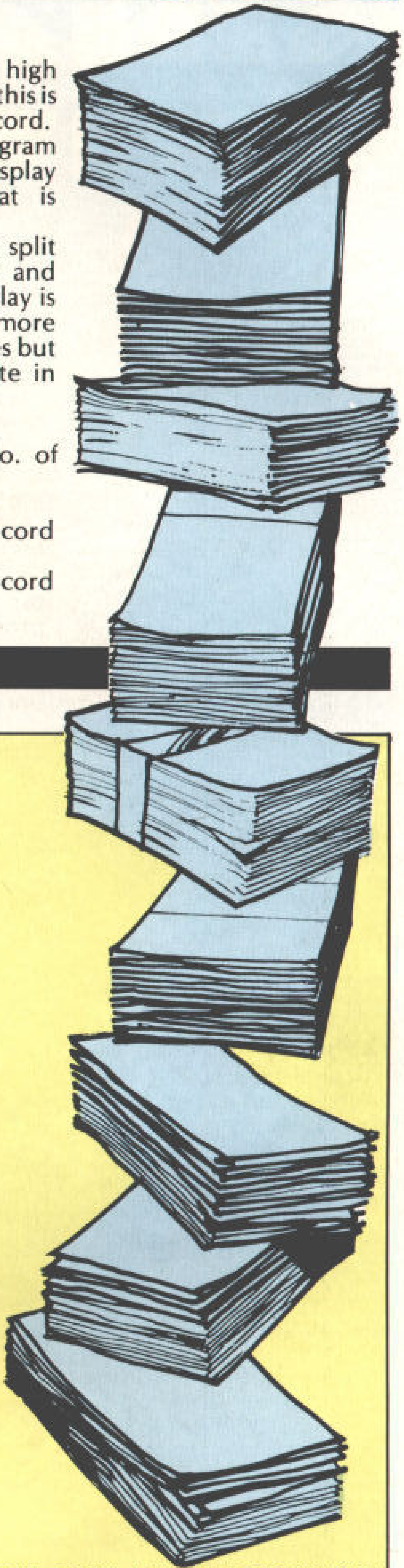
not possible to replace a relative file using "@0:. The only way to remove a relative file is by the scratch command.

Get running

When you run the first program you will notice that an error 'RECORD NOT PRESENT' appears. When creating a file this is quite normal and can be ignored.

What does the program do? **PROGRAM 1.** LINE 5 is the number of characters/record. To start with stick to 254 (this is the maximum possible).
LINE 6 is the number of records you want to create initially. Stick to about 100 for now.
LINE 10 opens the file and gives it a name. The ,L, part tells the disc that it is a relative file and the CHR$() tells the disc the record length.
LINE 40 is an easy way to split a record into a two byte location. N.B. the command AND.
LINE 50. The "p" stands for POSITION and this tells the disc drive to be ready to locate the read write head at the following track/sector. The CHR$()'s in this line are A: the channel being used (2 in this case) B: This is the low byte of the track/sector address. C: this the high byte of the address. D: this is location within the record. After running the program and watching the display you should see what is happening.
PROGRAM 2. This is split into two parts, Read and Write. The screen display is simple and needs more string handling routines but these are easy to write in later.

VARIABLES USED
RM = 1 Maximum No. of records.
RN = Record number.
RH = High byte of record number.
RL = Low byte or record number.



Program 2 Listing

```
READY.

1 PRINT"■":REM ***PROGRAM 2****
2 REM*****MAX REC= 668
3 POKE53280,0:POKE53281,0:PRINT"■"
20 INPUT"■■ READ OR WRITE R/W";Q$
21 IFQ$="R"THEN 1000
22 IFQ$<>"W"THEN20
23 OPEN15,8,15:OPEN2,8,2,"0:REL FILE"
30 INPUT "■ ENTER NEW RECORD £";RN
31 INPUT"■ENTER MESSAGE 1>";X$(1)
32 INPUT"ENTER MESSAGE 2>";X$(2)
33 INPUT"ENTER MESSAGE 3>";X$(3)
34 INPUT"ENTER MESSAGE 4>";X$(4)
40 RH=INT(RN/256):RL=INT(RN AND255)
50 PRINT£15,"P"CHR$(2)CHR$(RL)CHR$(RH)CHR$(1)
51 Z$=","
60 PRINT£2,X$(1)Z$X$(2)Z$X$(3)Z$X$(4)Z$:S=ST
65 INPUT£15,A,B$,C,D:PRINT"■■}ERROR IS = TO ";A:
66 PRINT"■■PRESS ANY KEY TO CONT."
67 GETQ$:IFQ$=""THEN67
260 CLOSE2:CLOSE 15:GOTO20
1000 OPEN15,8,15:OPEN2,8,2,"0:REL FILE"
1010 INPUT"■ RECORD NO. ";RN
1020 RH=INT(RN/256):RL=INT(RN AND 255)
1030 PRINT£15,"P"CHR$(2)CHR$(RL)CHR$(RH)CHR$(1)
1040 INPUT£2,X$(1),X$(2),X$(3),X$(4)
1050 INPUT£15,A,B$,C,D:IF A=50THENPRINT"RECORD NOT PRESENT TRY AGAIN":GOTO1000
1060 PRINT"■■}RECORD £  ";RN;" = "
1065 PRINT"■****************************************"
1070 PRINT"■";"1> ";X$(1):PRINT"2> ";X$(2):PRINT"3> ";X$(3):PRINT"4> "; X$(4)
1075 PRINT"■****************************************"
1076 PRINT"PRESS ANY KEY TO CONT."
1080 GETQ$:IFQ$=""THEN1080
1090 CLOSE15:CLOSE2:GOTO20

READY.
```


QUICKSILVA'D!
QUICKSILVA
All titles available from Quicksilva Mail Order, P.O. Box 6, Wimborne Dorset BA21 7PY.
GATE CRASHER
SPECTRUM 48K £6.95 BBC/ELECTRON £6.95
EPYX SUMMER GAMES
COMMODORE 64 TWIN CASSETTE PACK £14.95 DISK £19.99
GAMES 84
SPECTRUM 48K LIMITED EDITION SUPPLIED IN A BOX £9.95
TRASHMAN
COMMODORE 64 £7.95
TRAFFIC
TRAFFIC QUICKSILVA
COMMODORE 64 £7.95
BOOGABOO
BUGABOO QUICKSILVA
COMMODORE 64 £7.95 SPECTRUM 48K £6.95
FRED
FRED SPECTRUM GAMES FROM QUICKSILVA
COMMODORE 64 £7.95 SPECTRUM 48K £6.95
ANT ATTACK
ANT ATTACK SOFTSOLID 3D FROM QUICKSILVA
COMMODORE 64 £8.95 SPECTRUM 48K £6.95
QED
QUICKSILVA EDUTAINMENT DIMENSION
AVAILABLE SOON
SEND S.A.E. FOR CATALOGUE AND DETAILS
OF OUR PROGRAMS FOR BBC, ELECTRON
ORIC/ATMOS, ATARI, MSX, ZX-81, VIC
HAVE YOU JOINED THE
GAME LORDS YET
WARNING: These programs are sold according to QUICKSILVA Ltd's terms of trade and conditions of sale, copies of which are available on request.
GSH
Available at WHSMITH
Boots
WOOLWORTH
John Menzies
and leading multiple retailers and specialist computer stores.

DATA STATEMENTS

Great reading

TWO WELL-KNOWN AND established book publishers Corgi and Addison-Wesley, have joined together to produce a series of computer books and in October they are due to release three books for Commodore owners; all written by Brett Hale, the titles are Arcade Games For Your VIC 20, Arcade Games For Your Commodore 64 and More Arcade Games For Your Commodore 64. So keep your eyes on those book shelves!

Buying Powers

Purchase leger is a new business program available for the Commodore 64 from Kemp Limited. It is a high quality ledger accounting program for recording and analysing purchases and includes nominal analysis and will account for VAT. Ten menu options are available; the program can have up to 100 creditor accounts and up to 50 nominal accounts and can deal with up to 640 entries each month. It has a full range of printout and can also be used with Centronics interfaces. It is disc and tape compatible and can be transferred to disc without difficulty. The retail price is £27.95 on cassette and £29.95 on disc and comes with a 16 page manual. The program can be obtained from good computer shops or from Kemp Limited by mail order. The price includes VAT and postage and packing within Europe. Further information can be obtained by contacting Kemp Limited at 43 Muswell Hill, London N10 3PN, tel: 01-444-5499.

New stuff from PSS

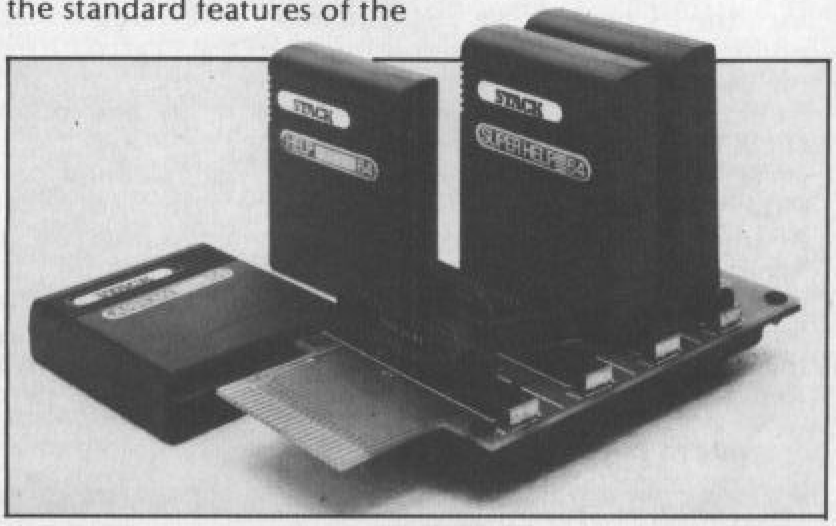
MIDWAY, FOR THE Commodore 64, is a diversion from other PSS products, being a strategic war game based on the World War Two clash between Japan and the U.S. The program has been written by a 'wargamer', who has ensured that the Japanese strategy, (the player takes the role of the U.S.), is the same as that in the real battle. There are three levels, making it suitable for beginners or experts. Midway will be available on disc or cassette at £12.95 and £9.95 respectively.

Getting Stacked

STACK COMPUTERS HAVE dropped the price of their programmers aid cartridge for the Commodore 64. Their cartridges which provide everything from extensions to BASIC, through fast tape operation and a two pass assembler, range in price from £25.00 upwards. When combined with their unique 4 slot motherboard, which allows use of programmers aid and games cartridges without powering down, Stack give the standard features of the 64 a very powerful boost.

For those of us with smaller budgets, the Stack 100 range now includes a cassette based compiler. Priced at £14.95 inc. VAT, the end-user can now achieve most professional results with his own BASIC programs.

More details are available from Stack Computer Services Ltd, 290-298 Derby Road, Bootle, Liverpool L20 8LN.



Commodore 16

THE NEW COMMODORE 16 is the ideal introduction to home computing, offering a powerful 16K RAM, full-size professional keyboard, 121 colours for high-quality graphics and sophisticated sound capabilities. Designed with the first-time user in mind, the Commodore 16 will be sold as a complete 'Starter Pack' comprising: computer, cassette unit, Introduction to BASIC Part 1 and four recreational programs — everything needed by the beginner — for £129.99. "The Commodore 16 is the perfect point of entry for anyone interested in serious home computing", said Howard Stanforth, UK General Manager of Commodore Business Machines. "We've packaged it in a Starter Pack because the VIC-20 has already proved that that's what the public wants and we believe that in this form it offers the best deal anywhere for programming, education or games." The new model features 32K ROM, advanced BASIC 3.5 with over 75 commands, including full graphics plotting and program editing. A unique HELP key is a particularly useful asset when learning to program, as it highlights errors in lines of programming right down to individual instructions. With 16K RAM standard, the Commodore 16 offers a full 12K RAM to the user for BASIC programs, making it a powerful machine for the games enthusiast on the one hand and the programmer or educational user on the other. The computer is equipped with: two joystick ports (to permit use of the new advanced Commodore joysticks); ROM cartridge and parallel disk drive port; Commodore 1531 Cassette Unit interface port; and a Commodore serial port offering access to a wide range of Commodore peripherals.

Spritely, spritely

GO-SPRITE IS A VERSATILE, easy-to-use sprite editor which makes full use of the extensive facilities on the popular Commodore 64, available from Mirrorsoft.

Among the many features of this Mirrorsoft machine stretcher is the fact that all program facilities can be operated by joystick alone. The Program also features light pen and keyboard control options.

Go-Sprite gives the user ease of use with icon driven commands — on-screen symbols give one-touch commands for all program features, simple animation of up to 32 sprites with 255 frames, easy handling of multi-coloured sprites, overlays of up to seven layers, user definable keyboard, sprite data files on disc and tape with data display option, and sprite editor.

Go-Sprite was written for Mirrorsoft by Bright Green Software.

DATA STATEMENTS



Great Eight

ACTIVISION IS NOW producing games cassettes for the Commodore 64 home computer. First titles off the production line are Beamridge, Decathlon, H.E.R.O., Zenji, Toy Bizarre, River Raid, Pitfall I and Pitfall II. Of these titles, Beamridger, Decathlon, River Raid and Pitfall I are new versions especially adapted and enhanced for the Commodore 64, of Activision's top-selling 1983 titles. Pitfall II and H.E.R.O. are Commodore 64 versions of Spring 1984 releases which are already high in the charts, and Zenji and Toy Bizarre are brand new Summer 1984 titles which are being launched simultaneously for the Commodore 64 and other systems.

Zenji is claimed to introduce the software entertainment industry's first mystical game of intuition. Begin at the source and *feel* your way through the unknown, connecting path after path until you've mastered the mystery and your senses. Zenji, designed for Activision by Matthew Hubbard.

In Toy Bizarre, it's midnight in the toy factory. Peaceful. Quiet. All of a sudden the balloon valves open. Gangs of tyrannical toys begin bounding from level to level taking over the toy shop. Can you stop them *before they capture you.*
Toy Bizarre, designed for Activision by Mark Turmell.

All these new Activision cassettes for the Commodore 64 retail at £9.99 including VAT and are available from video games and home computer software outlets everywhere.

Gortek rides again

FOLLOWING THE SUCCESS of 'Gortek & the Microchips', Commodore have launched an educational package for the 64, 'Gortek & the Kryptobytes. In the new program, space hero Gortek, together with his Microchips, again demonstrates a new and unusual way of learning to program in Commodore BASIC.

The first program has alreay been translated into at least five languages and 'Kryptobytes' is designed by the same team of three schoolteachers from Southern England. It is accompanied by a full colour story book which contains a 'flight file' to teach additional programming skills and programs to type into the computer. An acetate sheet 'map' is used as an overlay in the book to help the student create his own designs and make it easier to POKE them onto the screen. The scenario of the story centres on the planet Xerat where the Kryptobytes have established a data centre to store information they have acquired from other planets. However, a mysterious force begins to extract the information. Gortek and his Microchips are called to the rescue and, with you, have to try to neutralise the force before the safety of the whole universe is jeopardised! Gortek and the Kryptobytes teaches students the construction of programs using the REM and LIST statements: The storage of data and the use of the READ command; the application of FOR and NEXT, both as a delay and in nested form for loops. In addition practice is given in simple program editing; the use of GET and INPUT, as well as a host of other programming features which take the student to an advanced level.

Gortek and the Kryptobytes is designed for 10 to 13 year olds, although it is suitable for younger children with parental assistance, while older children and even parents will find it informative and great fun to use.

Pets can microwrite

COMMODORE PET USERS can now communicate with the Microwriter — the portable hand-held word processor with a unique and extremely simple to use keyboard of just six keys. Microcomputer Services, an appointed Microwriting Centre, has developed the software program 'Speakeasy', which allows two-way transfer of text between PETs and Microwriters.

Now PET users can transfer text to their data discs for storage, merging of files or for printing out a convenient moment. Documents can also be retrieved from the PET and entered into the Microwriter's memory for reference, updating or amendment. The Microwriter can also be used in a networked environment.

An interface lead, enabling communication between the PET, which has IEEE connectors, and the Microwriter's in-built RS232 is available from MicroComputer Services.

Due to Microwriter's total portability, it is especially suited for any situation where text needs to be writen away from the offices — whilst travelling or for meeting notes, or even at home out of office hours. Within the office, Microwriting provides an invaluable method of producing written text without recourse to dictation. Neither do notes need to be handwritten and later transcribed by a typist. The Microwriter user has complete control over content, format and amendments, with material being written whenever and wherever convenient. Hardcopy printout can also be obtained directly from the Microwriter linked to a printer.

The 'Speakeasy' program is available from MicroComputer Services, priced at £140 for two way communications, and £70 for one way. The 'link' will also work with all Commodore models, and details are available from: Microwriter Ltd., 31 Southampton Row, London WC1, Tel: 01-831-6801.

Super, Supersoft

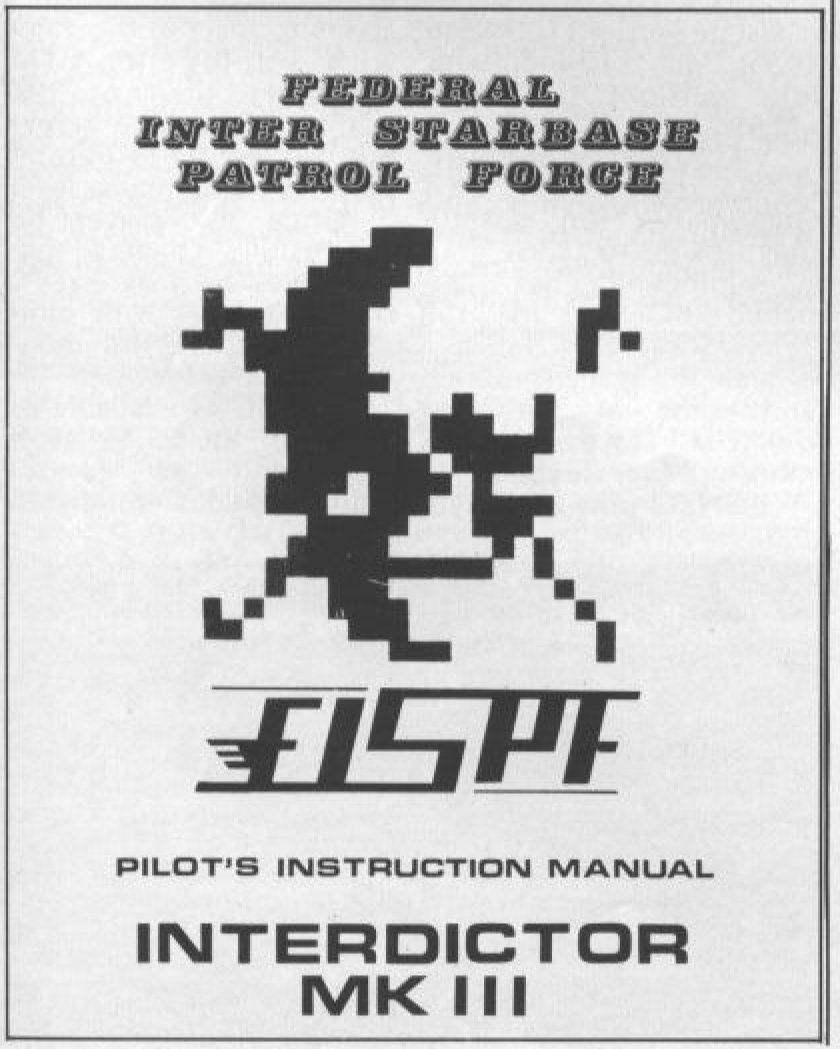
SUPERSOFT PRODUCE AN enormous range of programs and accessories for the Commodore micros. For example. . .

Available for the CBM 64 is a truly 3-dimensional spreadsheet program, Busicalc 3. This offers a variety of features over those included in the rest of the Busicalc range, including up to 999 rows and 200 columns, 3-dimensional formulae and user-defined functions, add or subtract sheets, fast FIND command, unique VU window function, link to Easy Script or Vizawrite and also to Chartpak 64, simple bar charts, sample files and utilities and variable column width. Busicalc works with all CBM printers, is available on disc only and costs £75 including VAT.

Still on the serious side of computing, Supersoft have taken on the marketing of Micro Applications' utility, Master, for the CBM 64. They have cut the price to just £69 including VAT, less than half the previous price.

Rather than produce yet another flight simulator for the Commodore 64 SUPERSOFT have gone one better with the release of INTERDICTOR PILOT, a space flight simulation! Written by a serving RAF officer, INTERDICTOR PILOT comes with a comprehensive 48-page manual which provides an insight into the latest technological developments including travel at the speed of light! In simulator mode the trainee pilot can find his way around the controls, and take part in simulated dogfights with alien craft. Many hours of simulator training are necessary before newly commissioned sub-lieutenants take part in real life sorties. INTERDICTOR PILOT is available from SUPERSOFT by post, or through dealers at £17.95 (including VAT). A disc version is also available at £19.95.

Why not find out more about Supersoft's range of products by contacting them at Winchester House, Canning Road, Wealdstone, Harrow, Middlesex HA3 7SJ. Tel: 01-861-1166.



Getting Educated

GET READY TO READ, THE first in a comprehensive suite of educational programs, featuring BJ the Bear, is now available from Commodore for use on the Commodore 64. Written by Dr. Richard Riding, Lecturer in Educational Psychology at the University of Birmingham, and Mrs Lillian Simmons, Headmistress at Moons Moat Nursery and First School in Redditch, Worcs, the program is available on cassette or disc and is accompanied by a BJ the Bear book and a parents' and teachers' manual.

Other programs in this series will include Get Ready to Think and Get Ready for Numbers. The Get Ready programs will be followed by Start to. . .and Continue to. . .for the same subjects but at increasingly advanced levels. The Get Ready programs are designed to introduce children of three-to-five years of age to the initial stages of reading. The A4 size, full colour "BJ the Bear" book is divided into four levels, each containing a different story about BJ, and is designed to be read to the child in order to prepare him for the learning activities of the program, all of which relate to the Bear's adventures. The book also contains simple pictures to match with letters of the alphabet, a guide to the forming of letters of the alphabet, and join-the-dots and colouring pages. A smiling BJ appears on the screen when the child has made the correct decision, and a frowning BJ appears when the child is incorrect.

The manual, which includes a set or progress charts, instructs the parent or teacher on how to prepare for teaching, how to use the programs, how to guide the child through the program, and how to grade their performance. "Get Ready to Read" is available from Commodore at £12.99

By Gum!

AFTER PACMAN, NOW welcome Plaqueman, hero of Tooth Invaders, one of Commodore's recent programs for the Commodore 64 home computer. Tooth Invaders demands enough computer gamesmanship to satisfy the most practised young tyro, whilst at the same time offering a worthwhile reminder on dental care — reassurance for Mum and Dad that computer games can have a useful message. As his name suggests, Plaqueman's mission is to eliminate all the plaque on tooth surfaces. He is guided by joystick and must collect a tootbrush and charge it with fluoride toothpaste to clean the teeth. He must then use dental floss for the gaps in between to complete the cleaning process.

But Plaqueman's enemy, DK, the Plaque Germ, gets ever more cunning and tooth decay sets in spontaneously, leaving only seconds for remedial treatment! The standard screen shows set of eight teeth — four up, four down — and there is a zoom screen to give a close up view of each tooth as Plaqueman goes to work. Plaqueman has three lives in each game and there are nine skill levels, plus a range of anti-plaque tricks any dentist would be proud of. When all the teeth are completely clean a fluoride rinse cloud descends and the game moves onto the next skill level.

The action is accompanied by music — for triumph or disaster, as appropriate — and there are special sound effects such as brushing noises and "decay alert" warning pips when plaque build-up becomes critical on untreated teeth. Available on cartridge, Tooth Invaders is priced at £9.99.

DATA STATEMENTS

Software Support

HANDIC SOFTWARE ARE obviously firm believers in the potential of the Commodore range of computers used in both the home and in business. Fairly recently they have produced a range of new programs specifically designed to boost the commercial applications potential of the 64, including such titles as CALC RESULT, DIARY 64, MON 64 (machine code monitors on cartridge), REL 64 (for such duties as activating burglar alarms, locks, telephones, central heating, lighting, etc),STAT 64 (for statistics and graphics displays), GRAF 64 (for studying complicated mathematical functions by their graphs), SUPERBOX 64 (an expansion board and parallel IEEE for the 64), PET-SWITCH and VIC SWITCH (a multi-user system for the VIC 20 and the 64).

Handic do in fact have a catalogue of all their products, so why not contact them and see what they have to offer? They are based at 5 Albert Road, Crowthorne, Berkshire RG11 7LT.

Getting Yourself Re-arranged

Based in Horsham Anagram Systems produce a range of Commodore software aimed at the small to medium sized company. The latest in their range of business accounting packages in Cash Book 64. Based on standard double entry book keeping practice, the system enables the user to maintain accounts to Trial Balance by the simple medium of recording Sales/Income, Purchases/Expenses, Assets and Liabilities. The Trial Balance may be produced for month end or year to date. The system will also record monthly budget figures and gives a budget to actual performance comparison. From a menu of only seven prompts, Cash Book 64 builds a complete accounting record including VAT summary, Audit Trail, Budget Summary and Performance as well as Trial Balance and Cash/Bank book. Cash book 64 costs £75 including VAT.

Going up-market somewhat, is the Integrated Accounting System running on Commodore 700 and 8000 series micros which has now been enhanced to include Sales Order Processing and full or partial Factoring. The IAS is based on standard double entry book keeping practice and comprises Sales, Purchase and Nominal Ledgers with Integrated Stock Control, full Purchase Order Processing and full Sales Order Processing. The full package allows some 200 book keeping operations to be conducted by means of just over 50 screen prompted functions which make it an ideal system for first time users or companies who do not have access to trained accountancy staff. Version 4 with Sales Order Processing and Factoring costs £1599 plus VAT.

Anagram Systems are at 60A Queen Street, Horsham, West Sussex.

Back To School

ASK HAVE ADAPTED SEVEN of their highly successful range of mathematics and literacy programs for the BBC Model B Micro to run on the Commodore 64. Available on cartridge at £9.99 in the Autumn is Number Painer for the 7-14 year age group. It is an arcade style game with 12 levels of difficulty to test your mental arithmetic. Three cassettes at £11.99 comprise Words, Words, Words; it links 10 different scenes together and the aim is to improve the spelling and vocabulary of children (7-10) who are learning to read. Number Chaser develops estimation skills through an exciting arcade game. This should be around at Christmas and is £9.99 as a cartridge.

Also available at the moment is Facemaker (building faces identikit style) at £4.99, Number Puzzler (noughts and crosses with a mathematical bent) at £4.99, Hide and Seek (for improving reading skills and practise short term memory retention) at £9.99 and Let's Count (to teach young children to count) at £9.99.

ASK specialise in educational software and the products are marketed by Commodore themselves.

Price War

THEY HAVE ANNOUNCED an entire range of price savings on many of their best selling games for the Commodore 64 and VIC 20. In some cases, Audiogenic has slashed prices by more than half. The price reductions include:
Commodore 64 software: Motor Mania (£8.95 to £5.95) Renaissance (£8.95 to £5.95), Grandmaster (Cassette version) (£17.95 to £8.95) and Grandmaster (disc version) (£17.95 to £12.95).

VIC 20 Software: Bonzo (£7.95 to £5.95) and Grandmaster (on cassette only) (£17.95 to £8.95).

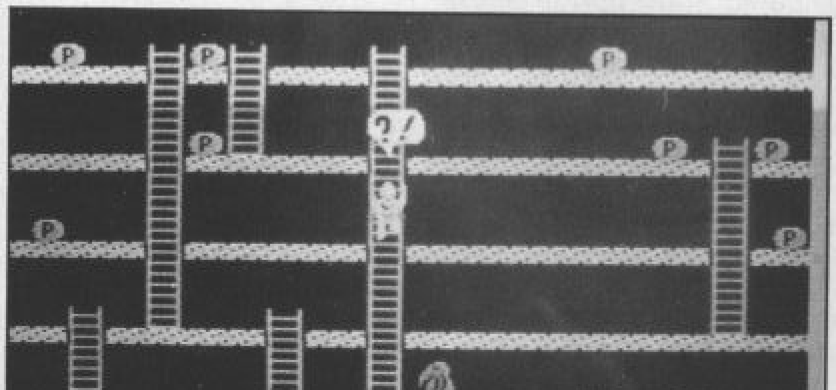
Audiogenic has launched BONZO for the Commodore 64 featuring brilliant full-screen graphics and some of the most devious and vicious monsters ever devised.

BONZO puts the player in the role of a workman whose job it is to collect boxes from different levels of the screen. The various levels are connected by ladders, of which several are available giving the player a choice of direction. Unfortunately, the monsters which guard the boxes can also climb.

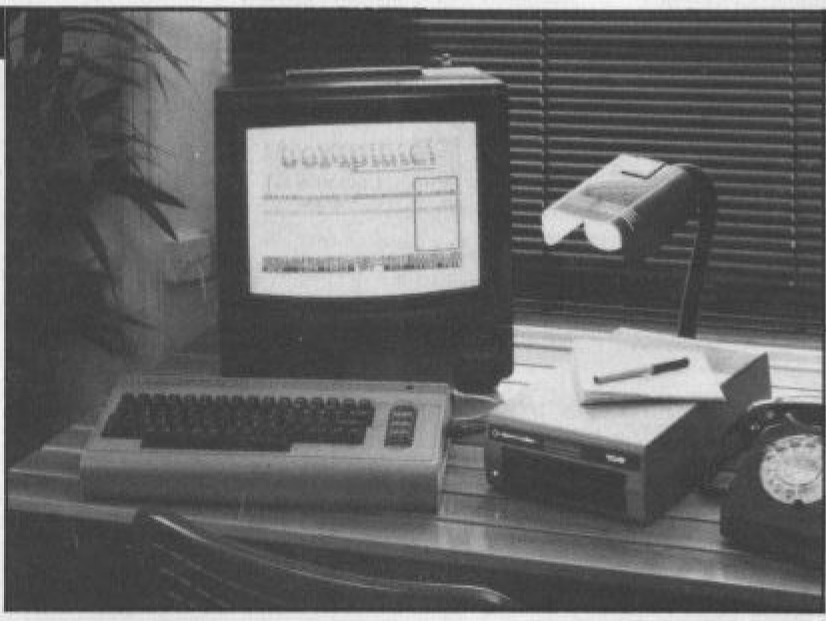
A unique factor of BONZO is in the monster's ability to react to the game with intelligence. The monsters do not just wander about the screen but actually try to trap the player on a particular level.

Once the player has successfully collected all the boxes, he progresses to the next level with more valuable boxes and more, even meaner monsters.

BONZO is available on cassette at £5.95 from Audiogenic or via the nationwide dealer network.



Compunet



COMPUNET IS THE NAME of a new service which will allow home computer owners to buy software, use financial and information services and even do everyday shopping direct from their own living room. They will be able to do all this using a simple telephone connection. The key item will be a small box called a Modem, which plugs into the computer and a telephone socket and allows the ordinary home computer to reach and communicate with the COMPUNET service, Commodore have developed the Modem especially for COMPUNET and an estimated 500,000 Commodore 64 owners, will have a chance to link up by the end of this year. COMPUNET will offer several key benefits that establish it as a premier service. Firstly, there is convenience. Users need only switch on and enter the nearest of 12 COMPUNET telephone numbers spread around the country. The modem's autodialler and software, contained in its built-in memory, will complete the connection in seconds. And once linked to the system, the software design means the user only ever has to use two or three keys to select and display material.

A second key advantage, and a remarkable development in itself; is the ability to offer reliable 2-way transfer of information and programmes, by cutting out the adverse effects of poor telephone lines. Thirdly, uploading as well as downloading is available. That is to say software and information can be transferred in both directions between the home computer and COMPUNET service. COMPUNET will offer quality software products at discount prices in an area called Software Park. Special features in the Modem and COMPUNET service prevent the software being run on any computer other than the one to which it was downloaded. By eliminating the risk of illegal copying the software protection facility makes the COMPUNET telesoftware service more attractive to suppliers of popular products. Another facility expected to interest computer clubs and hobbyists is an area of the service called 'The Jungle'. Subscribers can upload software and information they have created themselves directly into The jungle. Other subscribers can then download it on a free or chargeable basis. Seller's earnings, less a commission are credited to their COMPUNET account and once a quarter the balance is credited or debited to their Bank Account using a Direct Debiting Scheme. Other facilities include an electronic mailbox service, Commodore product information and alternative software which will alow the Modem to link with viewdata services.

The Modem costs £99.99 and, as an introductory offer, the first year's subscription to COMPUNET will be free to purchasers. Subscribers have free access in evenings and at weekends and can also use many parts of the service without charge. Later in the year, after an introductory period to fine-tune COMPUNET to user and service provider requirements, further retailing and information services will be brought on-stream. These will include catalogue shopping, home financial and insurance services, educational products and information facilities for the domestic environment. COMPUNET has been developed and introduced by Commodore Business Machines (UK) Limited and ADP Network Services Limited. COMPUNET will be operated on ADP computers and users will be able to link in via a local call to the ADP Network in twelve locations throughout the UK.

Note: For more information on modems, see the relevant article in this issue. Watch future issues of Your Commodore for reviews of modems themselves.

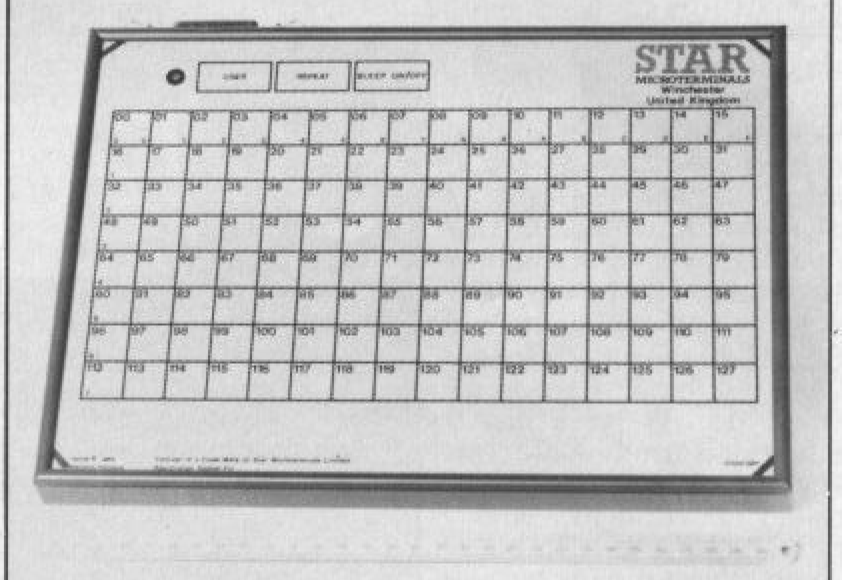
Tandata modems

TANDATA MARKETING have two 'smart' modems, the TM110 and TM120 together with Micropaks for the Commodore 64, Commodore PET and VIC 20. The TM 110 operates on 1200/75 baud V23 full duplex mode and is able to store and access data in its own CMOS RAM. It features an eight-telephone number store allowing up to 16 digits per number with a 32-digit reference comment for each number for ease of location and auto dial of stored numbers with auto recall of unobtained numbers. Up to eight separate ID numbers and passwords each with up to 16 alpha or numeric characters can be permanently and securely stored in the modem's own memory. Also the sometimes length log-on procedure is reduced to a single key function.

The TM 120 multi-rate modem includes all the features of the TM110 and in addition of 1200/75 baud full duplex, also offers 75/1200 baud full duplex and 1200/1200 half duplex to allow two micros to 'chat'. Both modems allow access to Prestel and its 300,000 plus pages of information, electronic mail and telex facilities, to Telecom Gold, Micronet and Prestel compatible private viewdata sstems. The TM120 costs £165 plus VAT. Software to allow a micro to act as a viewdata terminal is now available for the entire range of Commodore micros. For the Commodore 64, software is available both on disc (£32 plus VAT) and in ROM (£42 plus VAT) and gives full Prestel emulation with colour display and the inclusion of an off-line message editor. The micropak includes a plug-in card and comprehensive instruction manual. Software for the PET range is available on disc giving full Prestel emulation except for Reveal, Flash, Double Height and Separated Graphics. Display is in mono and off-line message editor is included; the pack comes with a plug-in card and a graphics chip. For the VIC 20, the software is on cassette with Prestel emulation as for the PET and costs £27 plus VAT.

More information is available from Tandata Marketing, Albert Road North, Malvern, Worcestershire tel: (06845) 68421.

DATA STATEMENTS



New concept

THE CONCEPT KEYBOARD is an original data input device offering many advantages over other keyboard systems. The user or programmer has complete flexibility in simply assigning the fixed codes generated from the CONCEPT keyboard matrix to characters, words, shapes, objects etc set out on an overlay, thereby giving the user the most efficient keyboard layout for a particular application. The underlying principle is a touch sensitive input selection array-sensitive over the entire matrix area — each touch cell producing its own unique output code.

Three versions of the CONCEPT keyboard are available:-
The A4 unit is 227 x 315 x 25 mm with a 16 x 8 matrix giving 128 touch cells each cell measuring 18 x 18 mm. The two A3 units are 315 x 435 x 25 mm. The A3-128 has a 16 x 8 matrix each cell meassuring 24 x 30 mm. With a 16 x 16 matrix the A3-256 has 256 touch cells each 24 x 15 mm.

The codes output from the CONCEPT keyboard are incrementing binary, commencing at cell 0 with Hex00 to Hex7F (FF) at cell 127 (255) compatible with the majority of common code systems — ASCII - ISO - EBCDIC for example. Both positive and negative strobes are available, making the interfacing to any computer system very simple. 2 cell Rollover with N cell lockout is standard on all CONCEPT keyboards. The standard output is 8 bit parallel, but 4 serial output options are available — Passive optically isolated 20mA current loop — EIA RS232C/V24 — RS422 — RS423. Each is supplied with a switch selectable baud rate generator, covering the range 50 to 9,600 bauds. The power requirement being very low is usually taken from the host computer. . .5 volts at 20mA.

The CONCEPT keyboard is easily interfaced into any computer system and a range of ready made-up interface leads are available for the majority of popular microcomputers including Commdore.

Prices start at £69.00 for the A4 CONCEPT keyboard and £149.00 for the A3-256.

ORC Attack

CREATIVE SPARKS HAS introduced a Commodore 64 version of its highly rated game *Orc Attack* which is already available for Spectrum and Atari computers.

In Orc Attack, you must defend your castle against the rampaging hordes of Orcs and their lethal crossbows. To defend yourself, you have a broadsword, rocks and, your ultimate weapon, vats of boiling oil. Orc Attack features four screens of action with excellent sound and original graphics, and is available on cassette for £7.95.

Also available from Creative Sparks for the Commodore 64 are two other games, *Black Hawk* and *Slurpy*. Black Hawk is a game of action and strategy, featuring the world's deadliest but fastest aircraft, and Slurpy is a greedy, cave dwelling character with the biggest appetite ever.

Creative Sparks games are distributed in the UK through THORN EMI Software Distributors and will be available in all principal retail outlets. Some products will also be available mail order.

Further information is available from THORN EMI Computer Software,
Creative Sparks,
Thomson House,
296 Farnborough Road,
Farnborough, Hants,
Tel: (0252) 543333

Definitely not rubbish

CERTAINLY A GAME TO have lent respectability to trash is the program from New Generation Software called Trashman. Originally for the Spectrum, it is currently being marketed in the USA for the Commodore 64 under the title of — wait for it — Garbage Gobbler. The only change is the name — the game itself is the same as that which was the subject of a recent competition for journalists, won by Computing Today's Editor, Peter Green. He came first playing the game on the Spectrum and won a weekend in Paris; he's just sitting waiting now for the chance to travel Stateside and throw down the gauntlet to our colonial cousins!

Mole in a hole

Wanted: Monty Mole is a new game from Gremlin Graphics for the Commodore 64 featuring a new micro character, Monty Mole. His adversary is the awesome Arthur (who'd have guessed?), safely ensconsed in his fortified castle and protected by a bunch of flying pickets. Monty tries to defeat Arthur by sneaking into the pit and snatching coal and as a final effort, steals the secret ballot papers and vote-casting scroll. Arthur and his flying pickets have a variety of means at their disposal to prevent Monty from succeeding. I personally haven't seen the game in action, and I'm all for innovative games that break away from the usual alien killing, but the idea seems to me to be in bad taste — the fact that this game was written by a miner's son only causes me more worry about the current mining dispute, not to mention the direction of computer games regardless of how good they may be technically. The game itself is priced at £7.95 and for every game sold, Gremlin Graphics will donate 5p to the Miners' Welfare Fund. Can we not opt out of this political levy?

CASH CONTROLLER
COMMODORE 64

HOME BUDGETING & BANKING SYSTEM
ON CBM 64 DISK AND CASSETTE

A professional style multi function Cash Controller program. In disk format you can load, and make an entry ... or update your existing records ... in just seconds. This easy to use program can handle up to 400 transactions, giving instant statements of your bank balance.

BANK ACCOUNT
- Statements on demand from any date to date.
- Holds up to 400 transactions.
- Standing order facility.
- All transactions can be automatically coded to allocate up to 16 budget headings.
- Search facility for any single item by description or amount.

HOME BUDGETING
- 16 budget headings, e.g. Gas, Rates, Car, Tax - you can choose the headings.
- Highlights under/over spending.

Full budget expenditure breakdown.
- Budget "re-think" facility.

LOAN/MORTGAGE CALCULATOR
All you ever wanted to know but didn't like to ask! Calculates:
- **Interest rates**
- **Payback periods**
- **Capital sum**
- **Period of loan**

- **PRINTOUT FACILITY USING STANDARD COMMODORE PRINTERS OR SEIKOSHA GP 100VC.**
- **FULL SECURITY WITH PERSONAL PASSWORD.**
- **SAVE DATA ONTO CASSETTE OR DISK USING 1541 DRIVE.**

DISK £14.95 TAPE £9.95 (inc. P&P)

CREDIT CARD HOTLINE 06286 63531 (24 HOURS)

Please send me:

________	Autocalc 64 DISK	at **£19.95**
________	Autocalc 64 TAPE	at **£14.95**
________	Cash Controller 64 DISK	at **£14.95**
________	Cash Controller 64 TAPE	at **£ 9.95**

(Overseas orders add £1.50 P&P).

I enclose Cash/Cheque/P.O. payable to Richard Shepherd Software Ltd. OR please debit my ACCESS/VISA card No.

Signature ____________________

NAME ____________ ADDRESS ____________

____

____

YC10-84

AUTOCALC 64
COMMODORE 64
THE SPREADSHEET IN A CLASS OF ITS OWN

TAPE £14.95 DISK £19.95

Q: Which spreadsheet is suitable for accountants, engineers, scientists and home users?
A: Autocalc 64 is ideal for any application involving extensive manipulation of data and formulae from financial planning to market research.

Q: Which spreadsheet offers an advanced level of formula handling?
A: Autocalc 64 copes easily with trignometrical functions, parenthesis and boolean logic as well as totalling and averaging.

Q: Which spreadsheet accepts complex conditional statements?
A: Autocalc 64 can handle statements as complex as IF a1<4,000 OR a1>8,000 AND a2 = 500 THEN b1=0.

Q: Which spreadsheet offers a flexible screen format?
A: Autocalc 64 allows you to select (i) column widths from 3 to 30 characters (ii) the number of rows/columns you need (iii) up to 2,000 cells of information (iv) text or numerical entries lined up to the right or the left, or a combination.

Q: Which spreadsheet offers a choice of numerical formats?
A: Autocalc 64 gives you a choice of (i) integers (ii) floating decimal point (iii) currency (iv) any combination of these.

Q: Which spreadsheet offers a full 'replicate' facility?
A: Autocalc 64 has an advanced replication function for transferring text, data, formulae or conditional statements from any cell (or block of cells) to any other(s) without monotonous retyping. A 'go to' facility will take the cursor instantly to any cell of your choice — saving time.

Q: Which spreadsheet is easy to use yet advanced in operation?
A: Autocalc 64 is designed to guide you — helpful error reports diagnose input or formulae errors. A full demonstration program and comprehensive instructions are included.

Q: Which spreadsheet is compatible with standard Commodore printers?
: Autocalc 64 gives you a printout facility using any of these printers: Commodore 1515, 1525, MPS 801, 1526, MCS 801, DPS 1101, Seikosha GP100VC.

Q: Which spreadsheet is 100% machine code for fast efficient responses, and offers a choice of saving to disk (using 1541 drive) or to tape using a C2N unit?
A: Autocalc 64 — as if you didn't know!

Q: Which spreadsheet sells at a realistic budget price?
A: Autocalc 64 costs just £14.95 on tape, £19.95 on disk inclusive of VAT and P&P.

Q: Where do I get one?
A: Ring us now on 06286 63531 (24 hours) to place your ACCESS or VISA card order, or complete the order form and send it to us today. (Prompt delivery promised). Autocalc 64 is available only direct from Richard Shepherd Software.

CREDIT CARD HOTLINE 06286 63531 (24 HOURS)

RICHARD SHEPHERD SOFTWARE
ELM HOUSE, 23-25 ELMSHOTT LANE, CIPPENHAM, SLOUGH, BERKS. TEL (06286) 63531

DOING IT YOURSELF

Business software can be expensive and doesn't always cater to all your needs. This series by Grahame Davies will give you an insight into how to write your own programs to suit your requirements.



THIS SERIES WILL ENABLE even the most experienced BASIC programmer to tackle the problem of storing and retrieving data for personal applications. It is based on the CBM 64 connected to a 1541 disc drive but all the techniques and logic used will be applicabel to all Commodore machines. Even if you only want to write games, the programming principals and discipline set out in this series will be of great use.

This series will not include a complete listing of a record keeping system but will provide utilities and techniques as well as explanations on how to successfully use disc files (and when to use each type). All the code has been deliberately left in BASIC but the routines may be converted into machine code if required. I suggest you read the articles and then practise some of the logic and ideas perhaps writing little test programs. Having satisfied yourself that you understand the principle (or at least have got your routines to work) then have a go at writing your own program. Just one other thing to remember — this series is designed for you to be able to write several systems, not just one, so if you find some of the explanations complicated or long winded do not be put off as you will really benefit in the end.

Setting out the programs

The first things to note are some of the do's and don'ts of BASIC programming. Firstly, bear in mind when writing that not only have you got to de-bug your program, but you may discover a bug in several months time, so make the code readable. Also remember that having got it working, you may want to modify it to store more data and if it is well written, this will be far easier to do. Do not try to put too many commands on one line. Having said that, it is unnecessary to put just one command on a line but with a little thought we can soon see some obvious guide lins. Here are a few examples:

```
10 rem your routine in here
20 rem and here
100 c = c + 1
200 if c<5 then 10
```

In this case, we alter the variable 'c' and want to immediately do a test on it to know whether to end or not. Having put the test on a new line, we have effectively separated the two. If we come back to this code and add a line 140 for example, although the code may still work the test has become further away from the increment making it more difficult to follow. Here is another:

```
100 for i = 1 to 10
200 if i =5 or i= 6then300
240 rem your routine in here
300 next
```

In this example we want to go round the loop ten times but ignoring numbers five and six. The test wants to be on the same line as the for. . next loop because there is no reason to have any code between them. Doing this will make the code safer and less prone to bugs if something is added.

Putting so many statements on one line saves memory and marginally speeds up the code but these advantages are insignificant and are far outweighted by the disadvantages — do not try to be memory efficient in this way.

We will now examine the REM command. This command is an essential programming tool but must be used sensibly:

```
100 nt = 0: rem initialise nt to zero
```

This is hopeless! The REM says exactly the same as the code. Much better would be:

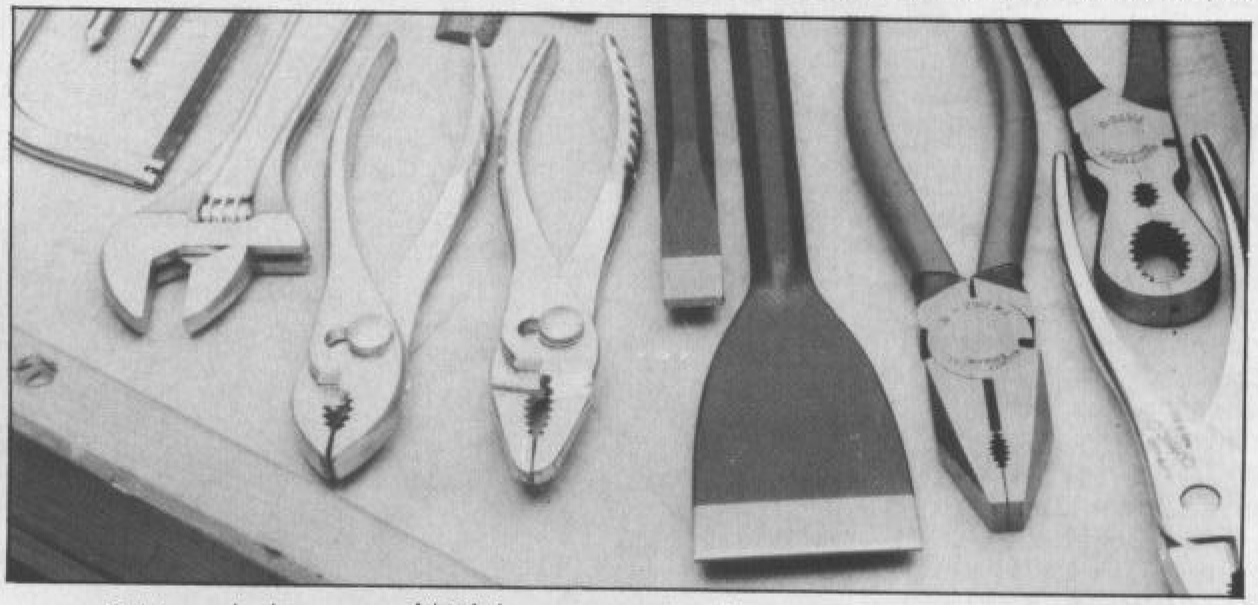
```
100 nt = 0 : rem number of times
```

This tells us what 'nt' is being used for. The code itself tells us that it is being set to zero, so why put it in a REM? Experience will show when to use REMs and what to put in them. For now, just remember that as you write a bit of code you understand it fully but in two months' time it will be harder to follow, so use some REMS telling you what variables are expected for the routine and what variables are returned. REMs do use memory and marginally slow the program — ignore these facts and use as many as you like. Try to format them neatly on the screen — leave spaces if necessary so that they may be easily read:

```
100 ex = 0: rem                          exchange rate
200 nt = 0: rem                          number of things
300 for i = 1 to 100 : next : rem        wait loop
```

Now let's decide what variable names to use. I advise having temporary variables which can be used to pass data to a subroutine and to be received back from subroutines. Also for common FOR. . .NEXT loops, try to use the same variables and avoid using any variable that you use in a subroutine. This is a common problem which causes FOR. . .NEXT loops to end suddenly or never end! For all the new flags etc, that is to say all new variables that are specific to the program being written and not in common subroutines, use two letters (or more) because this is more descriptive and easier to follow. 'AM' for amount is obviously quite clear but 'F9' for amount is obscure. Using more than two letters makes the code even clearer ('AMOUNT' for amount) but is unnecessary and has some disadvantages — it uses memory, it leads to more syntax errors as BASIC may discover a reserved word such as 'ON' in 'MONDAYS' and in any case BASIC only stores the first two characters of the variable name so the variables 'AMOUNT' and 'AMEND' are the same.

When you have completed your program, go back and read the above, then amend your program so that it is neat and easy to follow. In this way the benefits will become more obvious.



Getting framed

If your first program is successful and you want to write more, you will need a program frame containing your favourite tried and tested routines. I will call this program a NET. In your net you will need subroutines for screen handling, disk handling, error messages and all the little things that you only want to write once. After a while you may add to them or improve them making them more elaborate. By keeping these in a net you will have easy access to them and the subroutines will be in the same place in every program making each program easier to understand. As your routines get more advanced you will find them becoming like a set of advanced BASIC commands and you will really be setting up a tailor-made high level language (well higher than BASIC anyway).

The line numbers you use are a matter of personal preference but there are still some things to remember: do not start at line zero or one or ten etc — you can have line numbers up to 63999 so make good use of them. Any routines listed in this series will have line numbers between 1000 and 2999 but you may relocate them (changing all the relevant jumps etc) where you wish.

IS BIGGER BETTER?

Commodore have launched a new business machine with a full 128K of RAM and some interesting features. Will it appeal to businessmen? Simon Dismore investigates.



COMMODORE'S TOP-OF-the-range machine, the 8296-D, is an interesting mixture of the old and the new. Underneath the "melted ice-cream" casing is a system which is fully compatible with the earlier products in the 8000 range, but carries a full 128K of RAM, organised in blocks of 16K.

"Very nice", we thought, as our evaluation machine was unloaded from the container lorry (Commodore have not skimped on protective packaging). Our first question was "what can we do with all this memory?" — and here the answers were a little disappointing.

You might wish to use a computer either for developing your own

Memory blocks

Figure 1a

```
Bit 7:  ENABLE EXPANSION MEMORY
        0 disabled (default on power-up)
        1 enabled

Bit 6:  I/O PEEK THROUGH
        (addresses $E800 - $EFFF)
        0 disabled
        1 enabled

Bit 5:  SCREEN PEEK THROUGH
        0 disabled
        1 enabled

Bit 3:  SELECT MEMORY
        (addresses $C000 - $FFFF)
        0 block 2
        1 block 3

Bit 2:  SELECT MEMORY
        (addresses $8000 - $BFFF)
        0 block 0
        1 block 1

Bit 1:  WRITE PROTECT EXPANSION RAM
        (addresses $C000 - $FFFF)
        0 not protected
        1 protected (I/O is not protected
          if peek through is enabled)

Bit 0:  WRITE PROTECT EXPANSION RAM
        (addresses $8000 - $BFFF)
        0 not protected
        1 protected (screen is not protected
          if peek through is enabled)

CBM reserve bit 4 for future applications
```

```
CBM 8296 MEMORY MAPPING

0      MAIN RAM Row 0          0
16k    MAIN RAM Row 1       4000
32k                         8000

       EXPANSION RAM
       (block 0 or 1)

       OR...

32k    SCREEN               8000
36k    EPROM                9000
40k    EPROM                A000
44k    BASIC                B000

       EXPANSION RAM
       (block 2 or 3)

       OR...

48k    BASIC                C000
56k    EDITOR               E000
58k    I/O                  E800
60k    KERNEL               F000
64k                        10000
```

Figure 1b



programs, or for running other peoples' programs. 99% of purchasers fall into the latter category, and we were surprised to find that there was no software available that took advantage of the extra memory. One might expect that products might be in short supply for what is a relatively new machine — but the complete absence of appropriate programs is quite baffling. What can your friendly Commodore dealer demonstrate except the existing range of software, none of which was written with the 8296 in mind?!

First in the pipeline, said Commodore, would be Handic Software's CalcResult spreadsheet program. CalcResult for the 8296 is still a few weeks away from completion, but the authors very kindly offered to share their experiences with us, even to the extent of getting their programming manager to call us from Sweden.

They felt that the extra memory was "useful but not essential" for their sort of program. The benefit is that larger spreadsheets with more data can be manipulated in the memory of the system, while maintaining compatibility with data on earlier machines. Once CalcResult was up and running, their word processing package might follow. An interesting possibility would be the use of the expansion RAM as a "clipboard" to hold results while programs were being changed — the first step towards an integrated environment of the sort offered by AppleWorks and Lotus 1-2-3.

Handic also pointed out that they had to go into the machine with a soldering iron to strap the appropriate internal jumpers so that they could select between expansion memory and firmware under software control (once strapped, a USER call can be used to enable and disable any resident EPROMs or ROMs). Your dealer could probably do this for you if necessary. We would be interested to hear any other experiences in this area — perhaps Commodore should add switches which can be used to configure the system with less trouble.

Because the expansion memory is configured in fixed 16K blocks you would have to write your own memory allocation routines before attempting to access the memory for any sophisticated application (the clipboard idea would probably just use the memory as an ultra-fast disk drive). If you are converting a word processing or spreadsheet package, you will probably have existing routines which would work with little modification. Otherwise, you should be careful to design your allocation routines (which must sit permanently in the lower 32K main memory) at any early stage of development. One final word of advice from Handic — interrupts are tricky to program. Jumpers and interrupts aside, selecting memory blocks is easily performed by POKEing your selection into the memory control register at $FF00 (see Figure 1).

The 8296 also gives you four 'pages' of video memory, instead of the usual one. This could be very useful, and is much easier to program. Commodore suggests that you might use the additional screens for help screens and menus, though once again strapping might be required if you have fitted extra EPROMs.

The alternate screens are selected through a two

Screen handling

```
GENERALISED ROUTINE FOR SELECTING ONE OF THE FOUR CBM
SCREEN "PAGES" FOR DISPLAY

1000 rem Subroutine to select screen page
1010 rem On entry, variable 'sp' contains screen to select (range 0 to 3)
1020 if (sp>3) or (sp<0) then e$="SCREEN PAGE": goto 9000: rem Error trap
1030 poke 59520,12: poke 59521,(sp*4)
1040 return
```

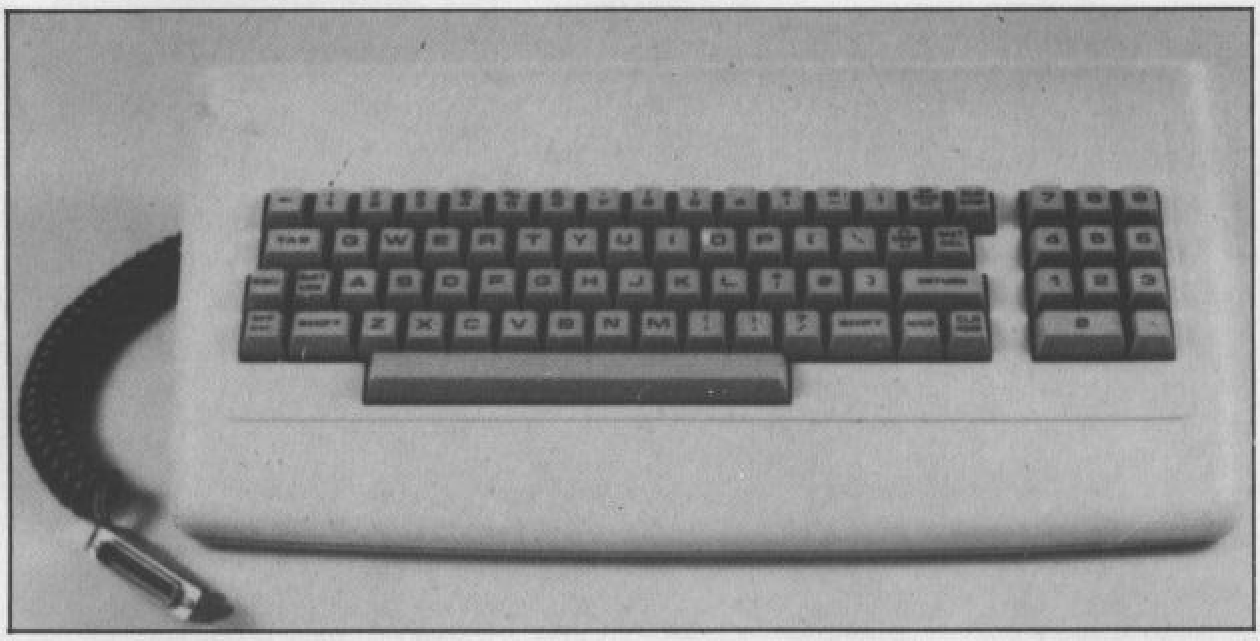
Figure 2

```
ROUTINE TO COPY ONE ENTIRE CBM SCREEN "PAGE" TO ANOTHER

1100 rem Subroutine to copy screen page
1110 rem On entry, variables select source 'ss' and destination 'sd' screens
1120 if (ss>3) or (ss<0) then e$="SOURCE PAGE": goto 9000: rem Error trap
1130 if (sd>3) or (sd<0) then e$="DESTINATION PAGE": goto 9000: rem Error trap
1140 sb=32768: rem Base of screen memory
1150 sg=2048: rem Gap between pages
1160 sl=1999: rem Length of a page, less one
1170 rem Calculate absolute address of source
1180 as=(sb+(ss*sg))
1190 rem Calculate relative address of destination
1200 ad=((sd-ss)*sg)
1210 rem Now copy
1220 for si=as to (as+sl)
1230   poke (si+ad),(peek(si))
1240 next si
1250 rem Copy completed
1260 return
```

Figure 3

line POKE sequence. Unfortunately, the PRINT statement in the resident basic (4.0) appears to work with only the default screen (Commodore number the screens from 1 to 4 in their documentation, but you will probably prefer to use a range from 0 to 3 for ease of programming — in either case, the default screen is the lowest number). This presumably means that you have to use the POKE statement to put characters onto any of the alternate screens. Unless we missed something in Commodore's (less than wonderful, as usual) documentation, this seems to be a major stumbling block to productive use of the extra screen memory.



Be that as it may, we did feel that the multi-page screens were worth the small amount of RAM invested in them. We have reproduced some simple routines which we wrote for screen handling. Figure 2 shows the POKE sequence for selecting between the four screens, while Figures 3 and 4 illustrate a full screen copy and a partial copy respectively. We learned (to our cost) that the 8296 does NOT take kindly to errors in POKE statements, so we have shown some ways in which errors can be detected and trapped (see Figure 5).

The most exciting application for multiple screens is almost certainly 'windowing'. You can try this with our partial screen program. The 8296 clears screen 0 on power-up but leaves the other screens with random values, so you would probably wish to clear them to blanks using a looping POKE 32 (space). The alternate screens are NOT cleared when you press reset, so any data you put in them is safe from everything except a power cut.

The resident BASIC is very slow (over 10 seconds to copy a screenful even with all remarks removed and multiple statements on a line), so you would almost certainly want to re-code any windowing routines into machine code once you have them working properly. Other amusing experiments include reversing rows and columns and simple text editing using the alternate screens as buffers.

If you are already a Commodore user and own a set of 8050 disk drives you might find the 8296 in its disk-less version was an interesting machine. It certainly doesn't seem worth replacing the 8096 with an 8296 just to get the benefits of additional memory for which no software is yet available.

If you don't already have a machine, should you consider the 8296-D? Weeellll. . .Commodore have produced a respectable but not exactly exciting product, which couldn't be regarded as 'state of the art' in any respect.

Disks: Single-sided, single-density 8050-compatible disks offer approximately 500K on each built-in drive, a figure which compares well with most other 5¼" disks. Maximum file size under DOS 4.0 is 180K, and there is room for up to 224 directory entries, both of which should be satisfactory for most purposes.

CPU: The 6502 certainly cannot compare with advanced 16-bit products

Screen handling

```
ROUTINE TO COPY PARTIAL CBM SCREEN TO ANOTHER
(USEFUL FOR "WINDOWING" SOFTWARE)

1300 rem Subroutine to copy a partial screen block
1310 rem On entry, variables select source 'ss' and destination 'sd' screens
1320 rem           variables 'x1' 'y1' 'x2' 'y2' give source co-ordinates
1330 rem           variables 'xd' 'yd' give destination starting point
1340 rem Validate 'ss' 'sd' as in lines 1120,1130
1350 rem Swap 'x1' etc to ensure top left, bottom right...
1360 t=x1: if x1>x2 then x1=x2: x2=t
1370 t=y1: if y1>y2 then y1=y2: y2=t
1380 rem Validate 'x1' etc...
1390 if (x1<0) or (x1>79) or (x2<0) or (x2>79) then e$="COLUMN": goto 9000
1400 if (y1<0) or (y1>24) or (y2<0) or (y2>24) then e$="ROW": goto 9000
1410 rem Validation of 'xd' 'yd' to ensure no illegal pokes
1420 if (xd<0) or (xd>79) or (yd<0) or (yd>79) then e$="START": goto 9000
1430 if ((xd+(x2-x1))>79) or ((yd+(y2-y1))>24) then e$="FINISH": goto 9000
1440 rem Calculate parameters (sb and sg defined in lines 1140,1150)
1450 as=(sb+(ss*sg))+(y1*80)+x1: rem Absolute source start address
1460 ad=(sb+(sd*sg))+(yd*80)+xd: rem Absolute destination start address
1470 rem Now copy line for line
1480 for yi=0 to (y2-y1)
1490   for xi=0 to (x2-x1)
1500     poke(ad+xi),(peek(as+xi))
1510   next xi
1520   as=as+80: ad=ad+80
1530 next yi
1540 rem Copy completed
1550 return
```

Figure 4



like the 68000 series. It runs slowly and has a limited addressing range. You might start to worry about its limitations after a while — it depends whether you expect that your applications will grow substantially within the lifetime of your purchase.

Screen: Contrast and character set are both good, and you can tilt and swivel the screen as required. However, we did find that the pronounced curvature of the screen and the high persistence phosphor took a bit of getting used to. This is simply a matter of individual preference, rather than a serious criticism.

The review machine tended to whine like an old monochrome TV — something which your dealer would presumably adjust before delivering the system.

Keyboard: The keyboard has a very 'springy' feel which would probably irritate touch typists at first. Keys must be depressed fully before a character is sent, so there is little risk of miskeying. We did feel that Commodore should have provided a full four-key cursor pad, rather than using shifts to generate cursor left and up from their opposites, given that this product is obviously intended for the serious business market.

We also thought that there should be an Enter key on the numeric keypad (generating a carriage return character). There is certainly room for these features on the detachable keyboard unit, which is one of the largest on the market.

As we go to press, Commodore are preparing plans for promoting the 8296-D from September onwards. The price of the system will be £1690 excluding VAT for the standard two-drive configuration. This price will include three "bundled" software packages covering the main applications areas. These consist of SuperScript, a word processor and spelling checker; The Manager, a database filing package; and CalcResult, a spreadsheet program. Bought separately, these would cost about £500. This makes the 8296-D reasonable value for money at the moment — though we are a little concerned that it could lose ground to some of the more up-to-date 16-bit products which will appear over the next six to nine months.

Handic's upgraded CalcResult will also appear in September, in a version which is compatible with all 8000 series machines (for further details contact their extremely helpful team on (0344-778800).

As to availability, Commodore tell us that 250 dealers throughout the country will be stocking the 8296. No new peripherals are being launched for the machine, but three compatible printers are available. Two are tractor feed dot matrix types with 60 cps/80 columns and 150 cps/132 columns respectively, while the third is a daisywheel friction feed printer working at 40 cps/136 columns. If you want the diskless version it will cost you £805 excluding VAT.

Finally, a message to Software Houses who are planning to use the expansion memory or the paged screens. Please drop us a line and tell us what you are up to, so that we can follow the progress of the 8296 in future issues!

Error trapping

```
COMMON ERROR TRAP ROUTINE TO DISPLAY MESSAGE ON PAGE 0

9000 rem Error trap with message in variable 'e$'
9010 rem Display diagnostic
9020 print "*** ILLEGAL";e$;" ***"
9030 rem Select page 0 for viewing
9040 sp=0: gosub 1000
9050 stop
```

Figure 5

Our fearless reviewer, Dave Crisp, gives you his general view of Superbase for the Commodore 64.

SUPERBASE OVERVIEW

ON MY COMMODORE 8096 I use the SILICON OFFICE and on my SX-64 I use SUPERBASE. If you have never used programs, like the above before, here is a brief description of what they are used for.

In their basic form they are best described as a prewritten database program; their power however comes when you want to cross reference information from one file to another, input new information and to generate a REPORT compiled from information from one or more files. Imagine for example three drawers of a filing cabinet. Each drawer contains a set of files,

eg, Drawer 1 - Names and addresses
Drawer 2 - Personal file (salary, car etc.)
Drawer 3 - Work record.

If the company wanted to send a standard letter to all employees who satisfy a particular set of conditions it would normally be a long and difficult job checking and cross checking information from the three drawers. With a program of the above type however a small sub-program could be written to examine all files in turn and create a list of all the people who meet the requirements. The list could then be used to personalise the stock letter which would have been created on a word processor. That is one use of a program like the above. Other uses are mailing lists, patient records, invoicing etc.

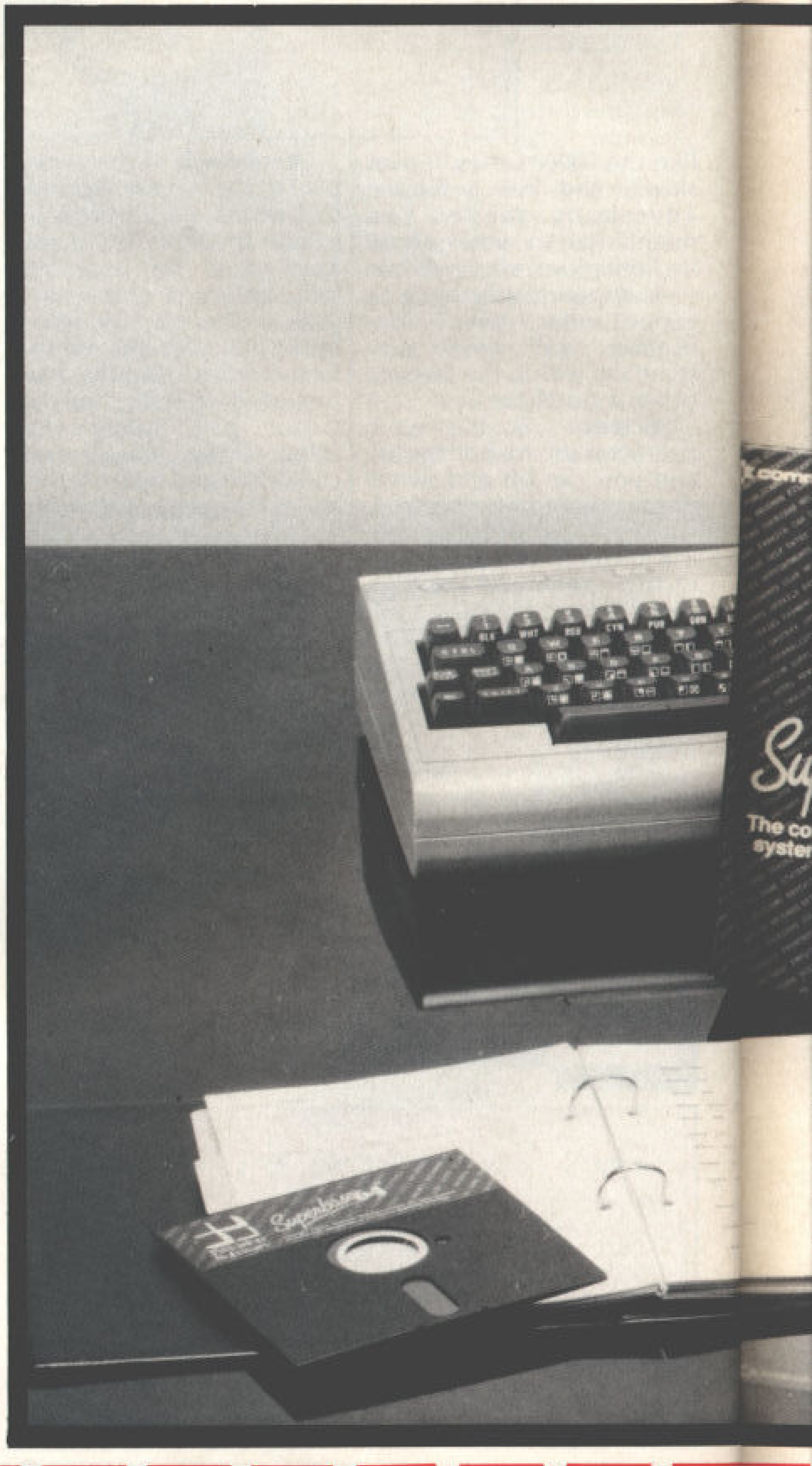
A program which is used to fulfil a specialised function through SUPERBASE and programs like it is called an APPLICATION and it is often possible to buy an applications package to run with the database management system.

Superbase

I will not try and describe how to use superbase as it is a very comprehensive program. I will just give an overview of it and a summary of how I have been able to use it. I had already learned to use the SILICON OFFICE and so when I first came to use SUPERBASE I was pleased to see that there were many similarities. Within a couple of weeks I had produced a stock control application linked to an electronic till drawer which could cope with about 700 stock lines and produce daily sales lists, low stock items, price lists, etc, etc. SUPERBASE can be used NAKED, by this I mean that it can be used as a menu driven database with all commands entered direct. By using the menu commands you soon get the idea of manipulating data in files and it is then a simple job of linking these direct commands together to form a program.

With discs

In use with a single disc drive it is possible to have up to 15 separate data files/disc and also any programs that you may have produced. The number of records in each file is in theory unlimited but it is of course restricted by the 170K restriction of the 1541 drive. It is possible and also easy to use two drives and using this system I have not yet found any limitations. With two

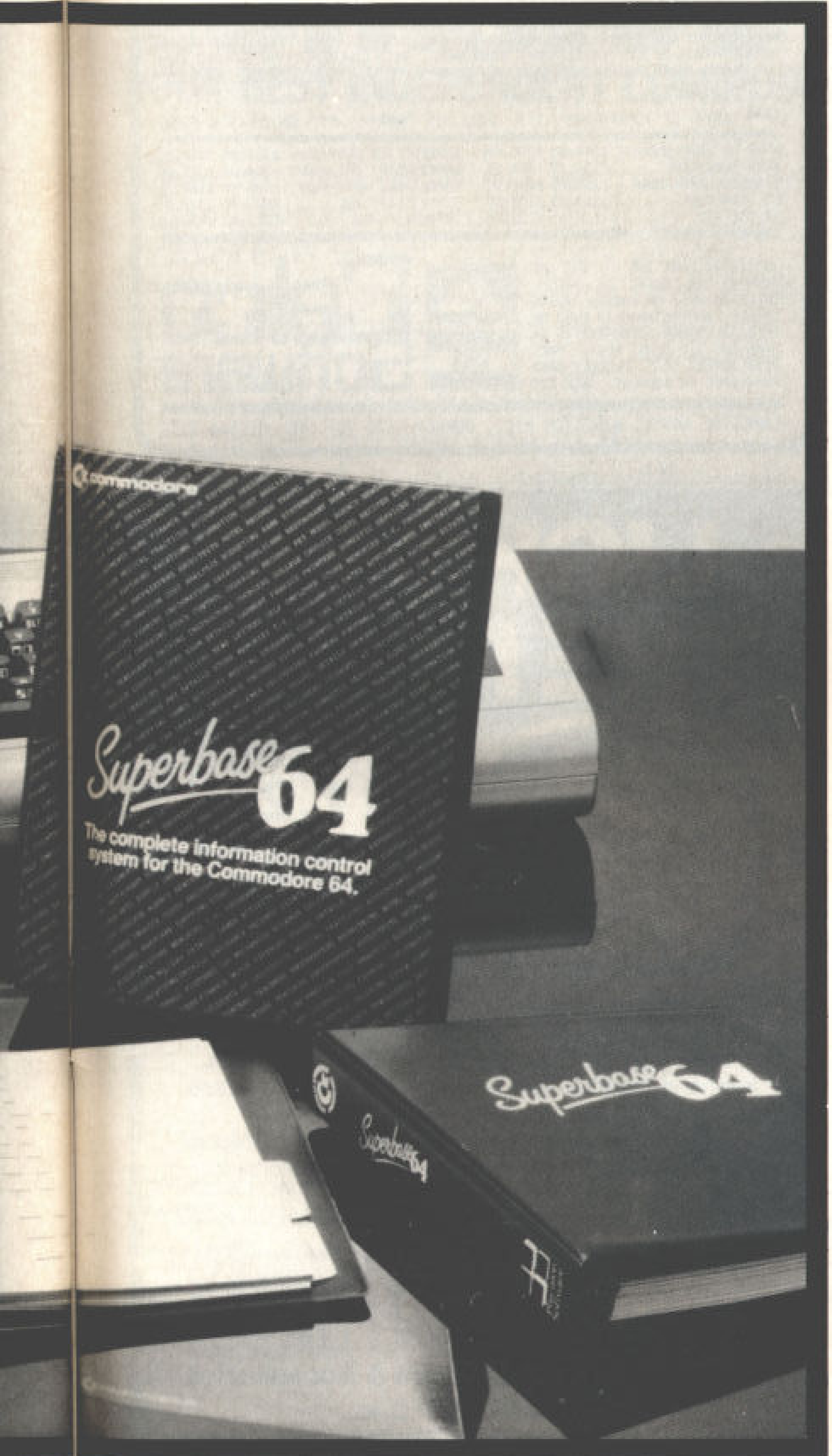


drives it is easy to make backup copies of your data. Initially I found I was not bothering to take copies of my data but the dreaded day came when with the help of a split cup of coffee about three weeks worth of data had gone. One of the interesting aspects of SUPERBASE is that it will link up to EASYSCRIPT and so personalised letters and documents are reasonably easy to create. I say reasonably easy to create because this is not a program that you would be able to just load and run — you would need to spend quite a few hours learning to link and manipulate files and to program effectively.

Basically speaking. . .

The language used by SUPERBASE is a mixture of BASIC plus some of its own commands. Its own commands can be entered in an abbreviated form or in a sentence, eg, to create a list of all records from a particular file you can write into a program 'display all records from file name' and to create a specialised list you can simply say 'find list where (town) = (town name)'.

This is the easy way to input commands but it is effective and once you have the idea you can input commands in an abbreviated form. The abbreviations are logical and are still meaningful when you go back to a program some time later.



There are built in pages called help screens which you can call up and these pages give you the syntax and use of most of the important commands available. On the whole it is user friendly program which is only limited in use by your imagination. Some of the important features of file manipulation are listed below with a brief explanation of their function.

1. BATCH. This allows the whole file to be updated automatically for instance if the VAT rate changed it would be possible to update each record with a simple list of commands.
2. FILE SIZE. It is possible to increase the length of a record without losing any data. This is a very useful feature and one which should be available on any good database.
3. FIND LIST. This allows you to create a specialised list of things which satisfy a given set of conditions. There can be as many conditions as you want and this created list can be used and destroyed or kept for future use.
4. DATE. This function is very useful. It enables you to set up conditions that must be met by a given date. Very useful on invoice/statement type applications.

It is possible if you have written an application with SUPERBASE to PROTECT it so that the program cannot be listed and also to password parts of the program to restrict unauthorised use, a feature which I have found useful to restrict access of personal details while allowing basic name address type information to be readily accessible. Screen colours can be changed easily and sound can be used to draw attention to important inputs. There is a built in software printer interface and so you should be able to use most of the common printers and switching from printer to screen is simplicity itself: if when writing a program you want to see the effects of a report on screen rather than on the printer it is just a matter of replacing the word PRINT with DISPLAY in the program. This can save pumping out reams of paper during debugging.

Great stuff

This is a program that I would not like to be without. It is versatile and probably the most useful of all my software. It costs about £100.00 but is considerably cheaper than its big brothers on larger machines. One thing that is important is when I had a problem with one of the programs PRECISION Software were very keen and quick to help. They do seem concerned with after sales service, something which is sadly lacking in many soft/hardware companies.

Dealers amongst you may be interested to know that there is a demonstration disc available which shows much of the potential of SUPERBASE.

If you are in business or run a club/society/have large collections/an untidy filing cabinet or just enjoy databases then this is the one.

Doesn't your 64 deserve a good book?

To make full use of the your Commodore 64's powerful capabilities, you need the help of good books. You need SAMS books – they'll keep you and your micro busy for hours.

Commodore 64 Programmer's Reference Guide

A bestselling book packed with professional tips to help you program your 64 successfully and creatively. Now at a new low price.

486 pages/ISBN 0 672 22056 3/**£9.95**

Commodore 64 Graphics and Sounds

Timothy Orr Knight

A combination of book and software to help you master the 64's powerful graphics and sound capabilities and use them in spectacular routines.

Book: 112 pages/ISBN 0 672 22278 7/**£6.95**
Book with software: ISBN 0 672 26186 3/
£15.95 + VAT

Commodore 64 BASIC Programs

Knight and LaBalt

A well illustrated collection of fun and practical programs, each of which is explained in full so that you learn how it works as you use it.

Book: 176 pages/ISBN 0 672 22171 3/**£7.95**
Book with software: ISBN 0 672 26171 5/
£13.50 + VAT

Learn BASIC Programming in 14 days on your Commodore 64

Gil Schechter

An easy to use and rapid method of learning how to program your 64, consisting of 14 chapters each of which can be covered in a day.

192 pages/ISBN 0 672 22279 5/**£10.50**

Available from all good booksellers.

SAMS books are distributed in the UK by Pitman Publishing, 128 Long Acre, London WC2E 9AN (Tel: 01 379 7383)

Pitman

Superbase 96

THE COMPLETE INFORMATION STORAGE AND CONTROL SYSTEM. SUPERBASE 96 TRANSFORMS COMMODORE MODEL 8096 OR 8296 INTO A FULL-FEATURED, PROFESSIONAL DATABASE SYSTEM! WITH UP TO 1100 CHARACTERS PER RECORD ON UP TO 4 SCREENS... AND UP TO 128 ITEMS PER RECORD, DEFINABLE AS KEY, TEXT, NUMERIC, CONSTANT, RESULT OR DATE... IN FILES OF UP TO 16M CHARACTERS! WITH SPREADSHEET AND CALCULATOR CAPABILITY, CALENDAR FUNCTIONS, MULTIPLE CRITERIA SORT AND SEARCH, EASY INPUT FROM WORDPROCESSOR/DATA FILES, BOTH MENU-DRIVEN AND PROGRAM OPTIONS, BATCH AND TRANSACTION PROCESSING, FULLY DEFINABLE OUTPUTS... SUPERBASE 96 IS ESSENTIAL IF YOU WANT THE MOST FROM YOUR COMMODORE! SUPPLIED ON 8050/8250 DISK, WITH EXCELLENT TUTORIAL/REFERENCE MANUAL. OUR PRICE ~~£454.25~~ £375.00!

Superscript II

SUPERSCRIPT II HAS ALL THE FEATURES YOU EXPECT FROM A PROFESSIONAL WORDPROCESSOR SYSTEM... AND THEN MORE! UNRIVALLED 240-COLUMN TEXT HANDLING, VARIABLE WIDTH SCREEN, FOUR-WAY SCROLLING OVER 2100 LINES, DOCUMENT LINKING FOR UNLIMITED CAPACITY, GLOBAL SEARCH AND REPLACE, FULL MEMORY CALCULATOR, ROW AND COLUMN ARITHMETIC, BACKGROUND PRINTING AND COMPLETE LETTER-QUALITY PRINT CONTROL MAKE SUPERSCRIPT THE ULTIMATE WORDPROCESSOR FOR THE CBM 8096/8296... AND IT SPELLS TOO! SUPPLIED ON 8050/8250 DISK. OUR PRICE ~~£454.25~~ £375.00!

Master

THE COMPLETE PROGRAM DEVELOPMENT PACKAGE FOR THE COMMODORE 8096/8296, USED BY BOTH NOVICE AND PROFESSIONAL PROGRAMMERS. MASTER ADDS 85 NEW COMMANDS TO BASIC INCLUDING 72K MEMORY MANAGEMENT, PROGRAMMER'S TOOLKIT, BUSINESS BASIC, KEYED DISK ACCESS, MULTIPLE SCREENS WITH USER-DEFINABLE INPUT ZONES, REPORT GENERATOR, 22-PLACE ARITHMETIC, DISK DATA COMPRESSION, DATE CONTROL, SCREEN PLOT, SCREEN DUMP, AND MORE... EVERYTHING YOU NEED FOR PROGRAMMING TO TOP PROFESSIONAL STANDARDS! WITH COMPREHENSIVE INDEXED USER MANUAL, QUICK REFERENCE CARD AND DEMOS. OUR PRICE ~~£339.25~~ £225.00!

*** WANT IT TOMORROW? *** CALL US TODAY! *** ON 01-546-7256

8032 TO 8096 UPGRADE	~~£346.00~~	£287.50	SUPEROFFICE 8096/8296	~~£799.25~~	£650.00
PM96 8096/8296	~~£114.53~~	£86.25	SUPERSCRIPT 4032/8032	~~£286.35~~	£225.00
VISICALC 8096/8296	~~£228.85~~	£195.50	SUPERSPELL 4032/8032	~~£172.50~~	£135.00
DTL COMPILER	~~£114.53~~	£109.00	SUPERSCRIPT 700	~~£517.50~~	£425.00
MASTER 700		£339.25	SUPERBASE 700	~~£517.50~~	£425.00

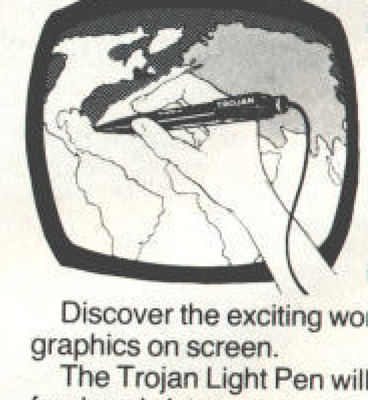
PRICES INCLUDE 15% VAT AND ARE CORRECT ON GOING TO PRESS. ORDER BY POST OR PHONE, USING CHEQUE, ACCESS/BARCLAY CARD OR OFFICIAL ORDER. DESPATCH IS BY SAME-DAY 1ST CLASS POST, PRE-PAID ORDERS FREE. PRODUCT DATA AVAILABLE ON REQUEST. REF A29

VISA

Calco Software

LAKESIDE HOUSE, KINGSTON HILL, SURREY, KT2 7QT. TEL 01-546-7256

LIGHT TO THE POINT



SPECTRUM 48K/COMMODORE 64

£17.25

DRAGON/TANDY

£11.50

inclusive

NO INTERFACE REQUIRED

Discover the exciting world of creating your own graphics on screen.

The Trojan Light Pen will draw boxes, circles, lines, freehand pictures, save and load pictures with full erase facility.

All in Hi-Res screen in any of 4 colours for the Dragon/Tandy, 8 colours for the Spectrum, and 16 colo''rs for the Commodore 64.

For educational or leisure use.

DEALER ENQUIRIES WELCOME

TROJAN

Micro Computer Software & Accessories

Send cheque/P.O. to.
TROJAN PRODUCTS
166, Derlwyn, Dunvant, Swansea SA2 7PF
Tel: (0792) 205491.

OH NO! SOLD OUT!

Your COMMODORE

YOUR BEST INDEPENDENT COMMODORE MAGAZINE

Whatever you do, don't let this happen to you. When you rush down to your newsagents to get your copy of the latest Your Commodore don't suffer the disappointment of being told they've none left.

Why not sit in the luxury of your own home and feel safe in the knowledge that your copies will come neatly wrapped winging their way through the post and find themselves popping through your letter box?

Whatever your interest in the Commodore range of computers and peripherals, you simply cannot afford to miss a single issue. The magazine is packed with news, reviews of the latest software, information about the books for sale, articles to help you with your programming, games to type in and play, useful routines to make your computing life easier, and all sorts of useful, informative and entertaining features. So whether your interest is purely in the latest games available for the VIC 20 or in reading a serious review of the most recent hardware for your Commodore, you must ensure that you read and inwardly digest every issue of Your Commodore.

And it's so easy to do! Just fill out the form below, write a cheque and send it off to the Subscriptions department. You know it makes sense!

SUBSCRIPTION ORDER FORM

Cut out and SEND TO:
YOUR COMMODORE
INFONET LTD., TIMES HSE,
179 THE MARLOWES,
HEMEL HEMPSTEAD,
HERTS, HP1 1BB

Please commence my personal subscription YOUR COMMODORE with the ....issue

SUBSCRIPTION RATES (tick as appropriate)

£13.40 for 12 issues UK ☐
£15.30 for 12 issues Overseas Surface ☐
£31.20 for 12 issues Overseas Air Mail ☐

I am enclosing my (delete as necessary) Cheque/Postal Order/International Money Order for£ ____________
(made payable to ASP Ltd)
OR
Debit my Access/Barclaycard★
(★delete as necessary)

VISA

☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Please use BLOCK CAPITALS and include post codes.

Name (Mr/Mrs/Miss)........................
(delete accordingly)

Address ....

..

Signature ____________ Date..........

**** COMMODORE 64 ****

R.B.S.
(RAPID BACKUP SYSTEM)

CONVERT YOUR SINGLE AND MULTIPART TAPE PROGRAMS TO TURBO-LOAD, FASTER THAN THE COMMODORE DISC DRIVE. EACH CONVERTED PROGRAM WILL LOAD INDEPENDENTLY OF R.B.S. NO POKES, NO SYS CALLS, NO USER KNOWLEDGE REQUIRED. LOAD "THE HOBBIT" IN 120 SECS. FOR PERSONAL USE ONLY.
CASSETTE £7.50 INC.

HYPERSAVE — 64

A VERSATILE AND FLEXIBLE FAST-LOADING UTILITY. 8 ADDITIONAL BASIC COMMANDS. SAVE/LOAD/VERIFY UP TO 48K CONTINUOUS RAM AT HYPERSPEED.
CASSETTE £7.50 INC.

BACK-UP — DELUXE

A POWERFUL AND USER FRIENDLY SERCURITY BACKUP UTILITY. LOAD/SAVE ALL PROGRAM PARTS CONTINOUSLY. INCORPORATES A SOPHISTICATED HEADER-READER. FOR PERSONAL USE ONLY.
CASSETTE £3.50 INC.

PRO-SPRITE

SPRITE LIBRARY. DESIGN, MANIPULATE, CATALOGUE MULTI AND SINGLE COLOUR SPRITES. CREATE DATA. JOYSTICK CONTROL. PACKED WITH FEATURES 100% MACHINE CODE. CASSETTE £7.50 INC. COMPLETE WITH YOUR FIRST SPRITE LIBRARY.

SPECIAL DISCOUNT: ANY TWO PROGRAMS £9.99. ALL FOUR £19.00 PLUS FREE! SECRETS! I/O GUIDE WITH EVERY ORDER.

CHEQUE/P.O. OR SAE TO:

DOSOFT
2 OAKMOOR AVENUE
BLACKPOOL FY2 OEE

***** FAST DESPATCH GUARANTEED ******

Superbase 64

TRANSFORMS THE COMMODORE 64 INTO A FULL FEATURED PROFESSIONAL DATABASE SYSTEM. WITH UP TO 1000 CHARACTERS PER RECORD ON UP TO 4 SCREENS... AND UP TO 128 ITEMS PER RECORD. DEFINABLE AS KEY, TEXT, NUMERIC, CONSTANT, RESULT OR DATE, IN FILES OF UP TO 16M CHARACTERS! SUPERBASE 64 EVEN HAS A SPREADSHEET AND CALCULATOR CAPABILITY, CALENDAR FUNCTIONS, EASY INPUT FROM WORDPROCESSOR OR DATA FILES, BOTH MENU-DRIVEN AND PROGRAM OPTIONS, SORTING/SEARCHING, FULLY DEFINABLE OUTPUTS... SUPERBASE 64 IS ESSENTIAL IF YOU WANT THE MOST FROM YOUR 64! SUPPLIED ON 1541 DISK WITH EXCELLENT TUTORIAL/REFERENCE MANUAL PLUS AUDIO LEARNING TAPE... AND THERE'S A WIDE CHOICE OF "GET-YOU-GOING" APPLICATION STARTER PACKS AT ONLY £9.95 EACH!
OUR PRICE ~~£99.95~~ £88.00

Jetpack

YOUR PROGRAMS WILL REALLY TAKE OFF WITH DTL JETPACK! FULLY COMPATIBLE WITH BASIC, JETPACK COMPILES BASIC PROGRAMS INTO MACHINE CODE, RUNNING UP TO 25 TIMES FASTER AND EVEN USING UP TO HALF AS MUCH MEMORY... AND IT ACCEPTS EXTENDED BASIC AND MACHINE CODE ROUTINES TOO!
OUR PRICE (DISK) ~~£39.95~~ £35.95

FIGARO 64

AT LAST, AN EASY-TO-USE YET POWERFUL NUMERIC DATABASE. WITH A FULL RANGE OF BUILT-IN STATISTICAL FUNCTIONS, PLUS FORECASTING, CALCULATION AND MIXED HI-RES/TEXT GRAPHICS. SUPPLIED ON DISK WITH CLEAR, COMPREHENSIVE MANUAL, FUNCTION-KEY OVERLAY, DEMO PROGRAM.
OUR PRICE ~~£86.25~~ £75.00

Vizawrite 64

VIZAWRITE 64 IS A HIGH-PERFORMANCE, LOW-COST WORD PROCESSOR, WITH ON-SCREEN FORMATTING, THAT TAKES FULL ADVANTAGE OF THE 64'S COLOUR, GRAPHICS AND MEMORY FEATURES... AND SUPPORTS VIRTUALLY ANY PRINTER! WITH A COMPREHENSIVE AND EASY-TO-FOLLOW USER REFERENCE MANUAL, VIZWRITE IS THE ULTIMATE PERSONAL COMPUTER WORD PROCESSOR! AVAILABLE ON CARTRIDGE (~~£89.95~~ £78), DISK (~~£79.95~~ £68) OR WITH VIZASPELL (~~£99.95~~ £85)

*** WANT IT TOMORROW? *** CALL US TODAY! *** ON 01-546-7256

VIZASPELL	(DISK)	~~£59.95~~	£49.95	MULTIPLAN (US)	(DISK)	~~£79.95~~	£67.95
VIZASTAR	(DISK)	~~£99.95~~	£89.00	PRACTICALC	(DISK)	~~£44.50~~	£39.95
DTL JETPACK	(TAPE)	~~£14.95~~	£13.95	PRACTICALC	(TAPE)	~~£39.95~~	£34.95
EASYSCRIPT	(DISK)	~~£75.00~~	£65.00	MASTER 64	(DISK)	~~£115.00~~	£64.95
EASYSPELL	(DISK)	~~£50.00~~	£41.95	THE TOOL	(CART)	~~£49.95~~	£39.95
SIMON'S BASIC	(CART)	~~£50.00~~	£41.95	HOMEBASE 1-4	(DISK)	......	£17.95
ASSEMBLER 64	(DISK)	~~£24.95~~	£21.95	PRINTLINK 64 INTERFACE		......	£29.95

PRICES INCLUDE 15% VAT AND ARE CORRECT ON GOING TO PRESS. ORDER BY POST OR PHONE, USING CHEQUE, ACCESS/BARCLAY CARD OR OFFICIAL ORDER. DESPATCH IS BY SAME-DAY 1ST CLASS POST. PRE-PAID ORDERS FREE. PRODUCT DATA AVAILABLE ON REQUEST. REF A30

VISA

Calco Software

LAKESIDE HOUSE, KINGSTON HILL, SURREY, KT2 7QT. TEL 01-546-7256

BUNTASOFT

GREAT ADVENTURES, STUPIDLY LOW PRICES!

KRAAL'S KINGDOM – MEDIEVAL TEXT ADVENTURE FOR COMMODORE 64 AND 16K VIC 20 (IN 2 16K EPISODES!)

★ ★ ★ ★ ★ FIVE STAR QUALITY – HOME COMPUTING WEEKLY £4·95

MAROONED – SPACEAGE TEXT ADVENTURE INC. SOUND EFFECTS. 16K VIC 20 ONLY £4·50

"GOOD VALUE..." – MICRO ADVENTURER

S.A.E FOR COMPLETE LIST.
1 CHIPPING HALL COTTS.
CHIPPING HERTS SG9 OPG



If an advertisement is wrong we're here to put it right.

If you see an advertisement in the press, in print, on posters or in the cinema which you find unacceptable, write to us at the address below.

The Advertising Standards Authority. ✓

ASA Ltd, Dept 3 Brook House, Torrington Place, London WC1E 7HN

AD INDEX

Anirog .... IBC
A P S 69
Audiogenic .... IFC
Calco 90
Honeyfold.................... 42
Llamasoft 23
Pitman Publishing 88
Quicksilva.................... 72
Richard Shepherd 79
Supersoft..................... OBC
Terminal Software 29
Trojan Products 88

ANIROG

NEW TURBO 64 GAMES



P.C. FUZZ



Aggro at closing time at the local. Getaway car screeches to a halt outside a bank. Ingenious methods employed by the Mafia to literally spirit away the loot from the High Street. Never fear - P.C. Fuzz is on patrol.

COMMODORE 64 £7.95

ZARGA MISSION



An action packed 3D scrolling arcade thriller by the author of Space Pilot.

COMMODORE 64 £7.95

Also available on Disk at £9.95

TRADE ENQUIRIES: **ANIROG SOFTWARE LTD. 29 WEST HILL DARTFORD KENT (0322) 92513/8**

MAIL ORDER: 8 HIGH STREET HORLEY SURREY 24 HOUR CREDIT CARD SALES HORLEY (02934) 6083

PAYMENT BY CHEQUE P.O. ACCESS/VISA 50p POSTAGE & PACKAGING

BUSICALC 3

- the sophisticated spreadsheet !

Easy to learn, easy to use - something that can't be said of many business programs. But it's true of all the programs in the BUSICALC series.

BUSICALC 3 can handle all sorts of jobs - budgets, expenditure analysis, stock lists, price lists, and product costing are just a few of the possibilities. Three-dimensional formulae automatically access data stored on disk, so that you can easily pull together information from several different sheets and summarise or manipulate it.

It's simple to transfer data to other programs such as Easy Script. And you can use virtually any printer with BUSICALC 3, whether dot matrix or daisy wheel, Commodore or non-Commodore.

For the CBM 64 and PET/CBM 4000 & 8000 series.

Available through dealers or from:
Supersoft, Winchester House, Canning Road, Harrow HA3 7SJ

Phone 01-861 1166 for more details and a free catalogue.

